MAKE | BUILD | HACK | CREATE

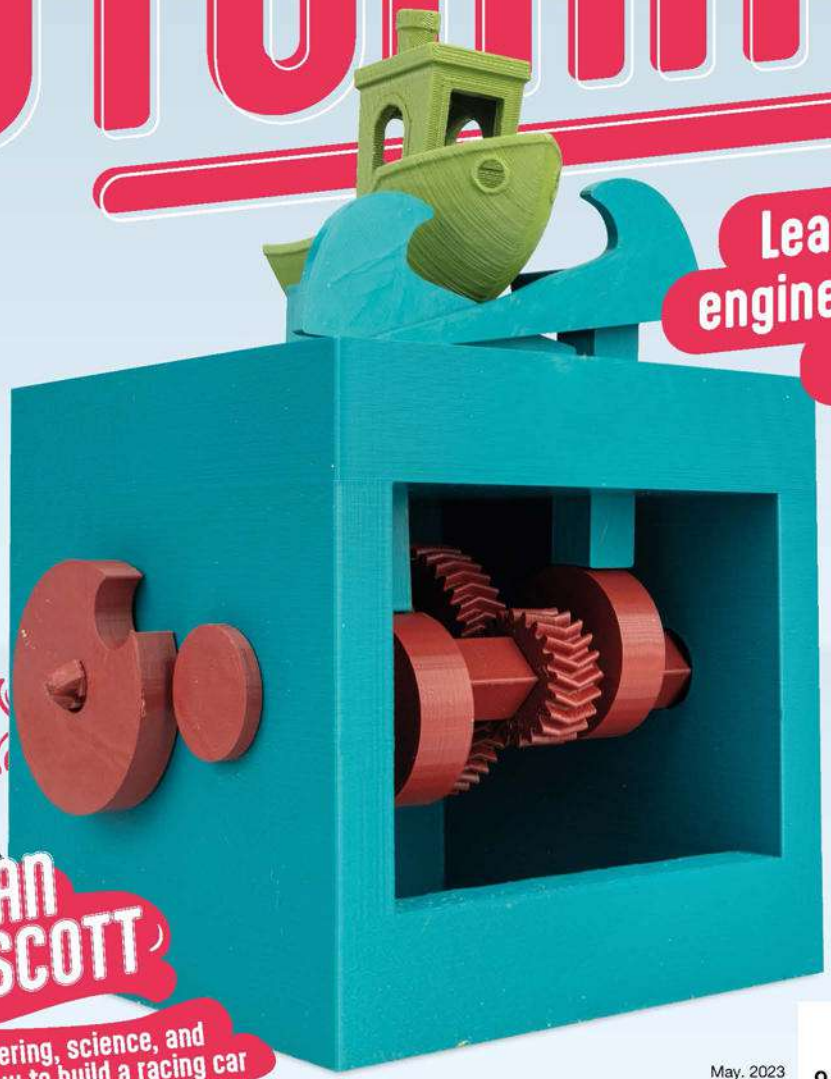
HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.CC | May 2023 | Issue #66

AUTOMATA

Learn mechanical engineering with a 3D printer



KICAD

Design and build custom PCBs



FRAN SCOTT

Engineering, science, and how to build a racing car

PAPER PLANES

The ultimate folded flying machine

May. 2023
Issue #66 £6

9 772515 514006 66

WILDLIFE **NYLON** SOLDER **CAT GPS**

Free eBook!



Download your copy from

hsmag.cc/freecadbook



Welcome to HackSpace magazine

When I was young, experimenting with mechanisms could be a really frustrating activity because things needed to fit together, and they seldom did. Gears scavenged from old toys were useless if they didn't mesh together, or if you didn't have a way of holding their axles in the right position. Some systems, such as Lego Technic, got you a bit closer, but I often found that I was missing the bit I needed to complete my machine. Thanks to 3D printers, those days are behind us. If I need a gear with 32 teeth and a module of 1, I can just pop those into a gear generator and have that part in under an hour. This lets us experiment far more freely than we could a few decades ago. In this issue, we'll be running some of those experiments, and all you need is a replica and a spool of PLA. Let's build some mechanisms!

Gears scavenged from old toys were useless if they didn't mesh together, or if you didn't have **a way of holding their axles in the right position**

BEN EVERARD
Editor ben.everard@raspberrypi.com

Got a comment, question, or thought about HackSpace magazine?

get in touch at **hsmag.cc/hello**

GET IN TOUCH

hackspace@raspberrypi.com
hackspacemag
hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play

PAGE 32
FREE PICO W
WHEN YOU SUBSCRIBE



EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Sub-Editors
David Higgs, Nicola King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Olivia Mitchell

Photography
Brian O'Halloran

CONTRIBUTORS

Marc de Vinck, Andrew Lewis, Sahas Chitlange, Jo Hinchliffe

PUBLISHING

Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.com

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6, The Enterprise Centre,
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE

To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

recycle

This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

HackSpace magazine is published by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents



72

06 SPARK

06 **Top Projects**
Wonderful builds to amaze and delight

16 **Objet 3d'art**
The world's biggest metal 3D printer

18 **Letters**
There's a lot of love for the Prusacaster

21 LENS

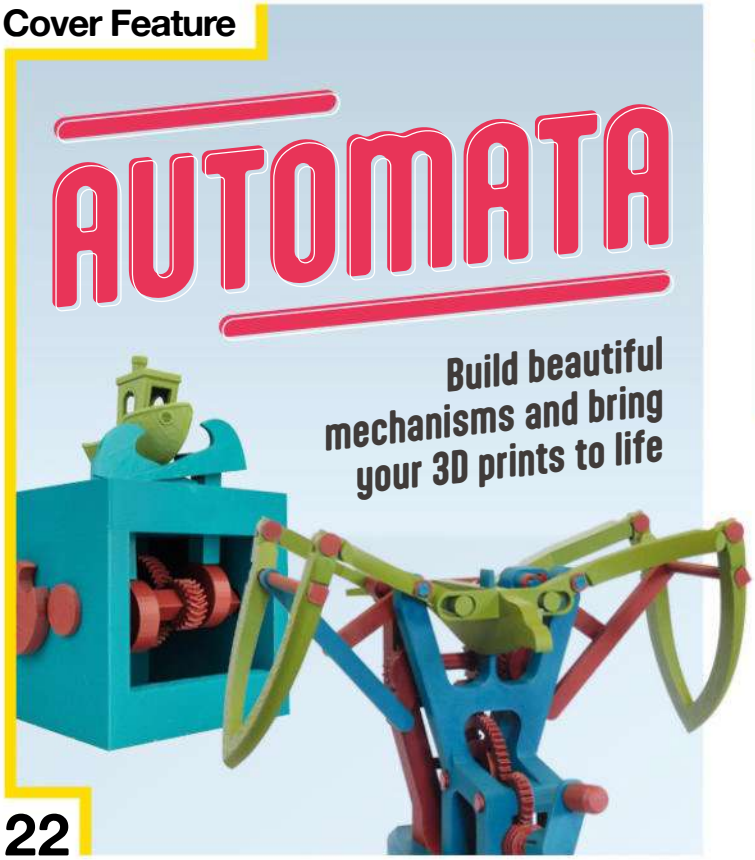
22 **Automata**
Mechanical engineering for 3D printing

34 **How I Made:** Cat tracker
Cats cannot be controlled, but they can be monitored

40 **Interview: Fran Scott**
On TV, perfectionism, and why messy is good

48 **In the workshop**
Monitor the creatures of the night

Cover Feature



22

Tutorial

Paper aeroplanes



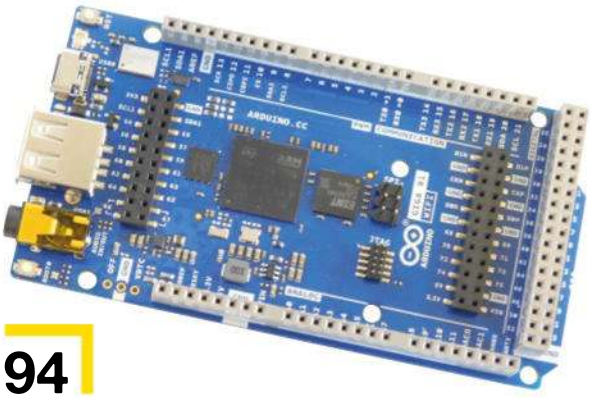
66 By far the cheapest way of learning to fly



06



34



94

Tutorial

Electric scooter



80 Add a Raspberry Pi Pico and a motor to your humble wheels

Interview

Fran Scott



40 Blowing things up for a living never looked like so much fun

53 FORGE

54 **SoM** KiCad
PCB design the open-source way

60 **Tutorial** Surface-mount soldering
Stick another circuit board in the oven

66 **Tutorial** Paper aeroplanes
Learn the complexities of flight with a sheet of paper

72 **Tutorial** Nylon
Printing with low-friction filament

78 **Tutorial** Hardware hacking
Resurrecting an Amstrad Z80

80 **Tutorial** Scooter
Electrify your ride with a Raspberry Pi Pico

85 FIELD TEST

86 **Best of Breed**
What can you actually get hold of these days?

92 **Review** Karakuri: How to make…
… Mechanical Models That Move

94 **Review** Arduino Giga
More connectors than you could possibly imagine

96 **Crowdfunding** Pocket Reform
An open laptop we can get on board with

Some of the tools and techniques shown in HackSpace Magazine are dangerous unless used with skill, experience and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. HackSpace Magazine is intended for an adult audience and some projects may be dangerous for children. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in HackSpace Magazine. Laws and regulations covering many of the topics in HackSpace Magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in HackSpace Magazine may go beyond. It is your responsibility to understand the manufacturer's limits. HackSpace magazine is published monthly by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS, United Kingdom. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Application to mail at Periodicals prices is pending at Williamsport, PA. Postmaster please send address changes to HackSpace magazine c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

REGULAR



Bubo

By **Kevin McAleer** Kevsrobots.com

Say hello to Bubo, the 3D-printed steampunk owl by Kevin McAleer. Bubo has a Raspberry Pi Camera Module 3 in one eye, and can recognise hand gestures to take photos and send the results to Twitter or to Mastodon.

Gesture recognition is provided by CVZone running on a Raspberry Pi, while each eye has a NeoPixel ring controlled by a Raspberry Pi Pico. Bubo can also blink, thanks to Kev's own 3D-printed mechanism and a servo for each eye.

That's the technical aspect: artistically Bubo is inspired by the clockwork robot from Ray Harryhausen's *Clash of the Titans*, Tik-Tok from the nightmarish *Return to Oz,* and the Nautilus from *20,000 Leagues Under the Sea.*



Left
Bubo was a hit at this year's Steampunk Weekend held in beautiful Whitby

REGULAR

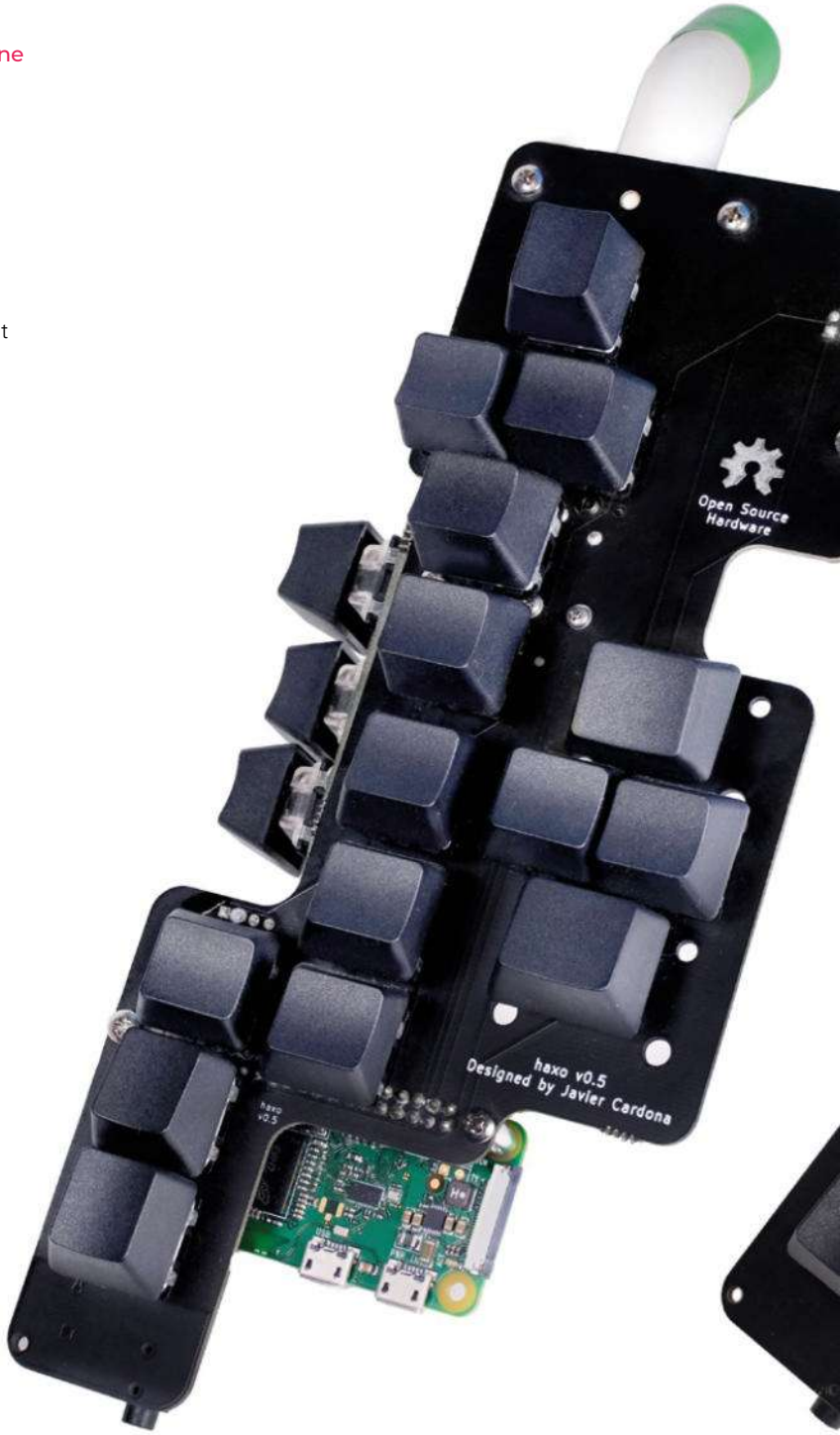
Haxophone

By **Javier Cardona** hsmag.cc/Haxophone

The Haxophone is an electronic take on the weird brass/woodwind chimera, the saxophone. Its maker, Javier Cardona, modestly reckons that it'll never be a substitute for the sound of a saxophone, but it is a lot more portable, and is likely to be a hell of a lot cheaper than a decent sax. It's also OSHWA-certified open-source hardware, so if it breaks, you can just 3D-print a replacement part.

The Haxophone is a Raspberry Pi HAT programmed in Rust, and will be appearing on **Crowdsupply.com** soon.





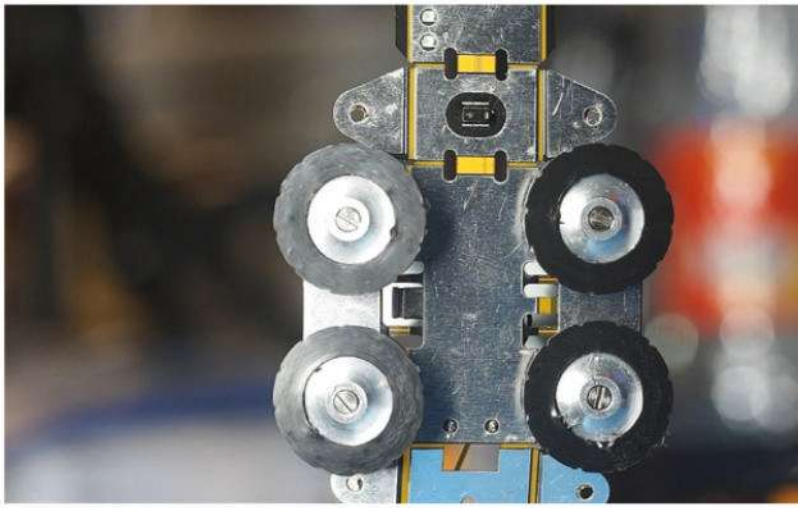


Above
Saxophones don't have a 3.5mm output jack for neighbour-friendly practice. This does!

Credit: Crowdsupply

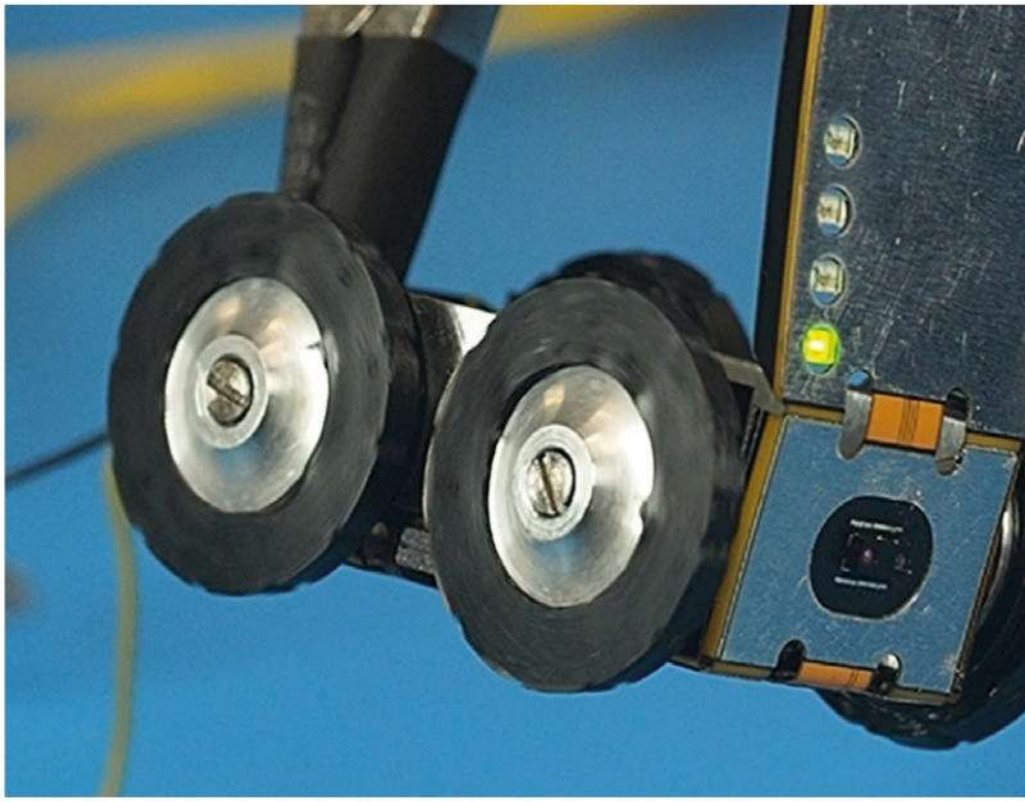
Folding rover

By **Carl Bugeja** carlbugeja.com



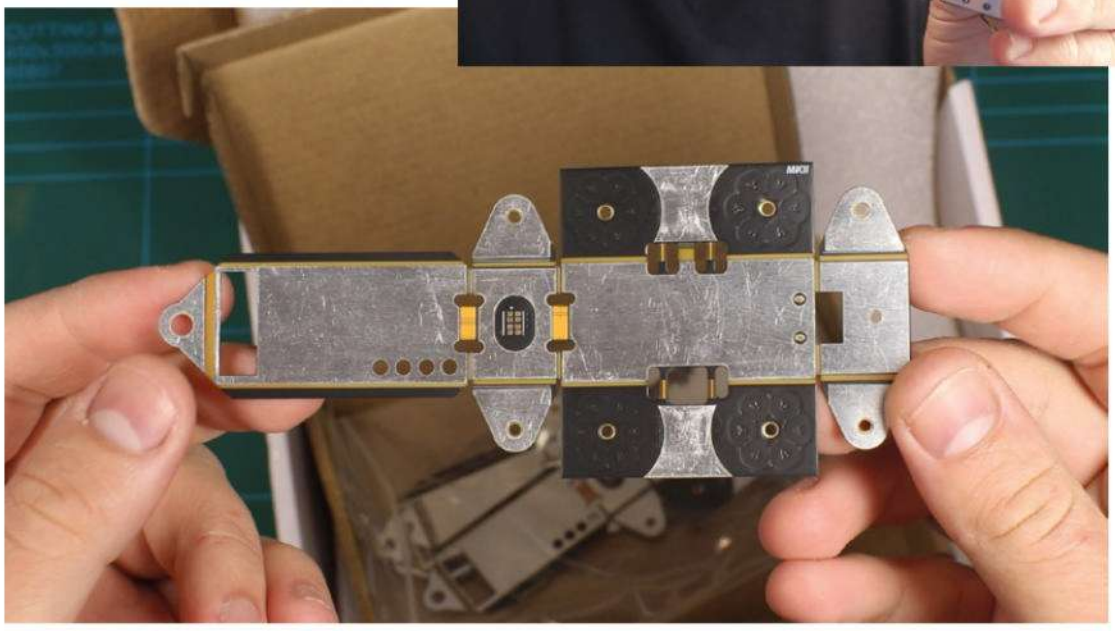
We've seen the flexible PCBs of electronics guru Carl Bugeja before – now here's a semi-rigid PCB that folds up to become a robot rover. Carl has incorporated another of his trademarks, the PCB motor, into the design, with a motor and a driver for each wheel. Control comes from an ESP32, which enables the user to connect to a Sony PlayStation controller via Bluetooth.

True to form, Carl's already working on the next version, ironing out the kinks that he discovered when he folded this up and tested it out. It's already working as a proof of concept though: check out the video on his site to see just how workable a folding PCB really is.

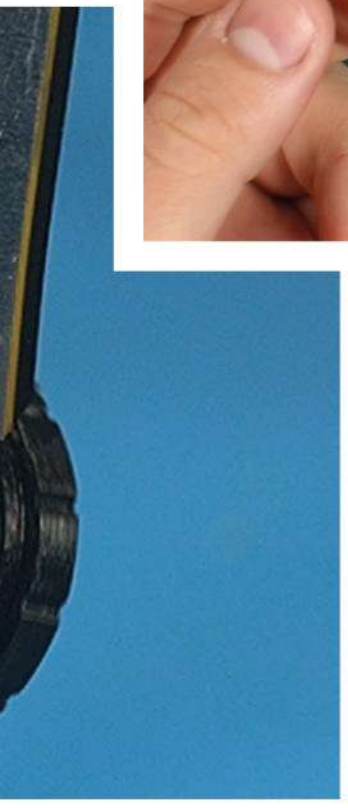


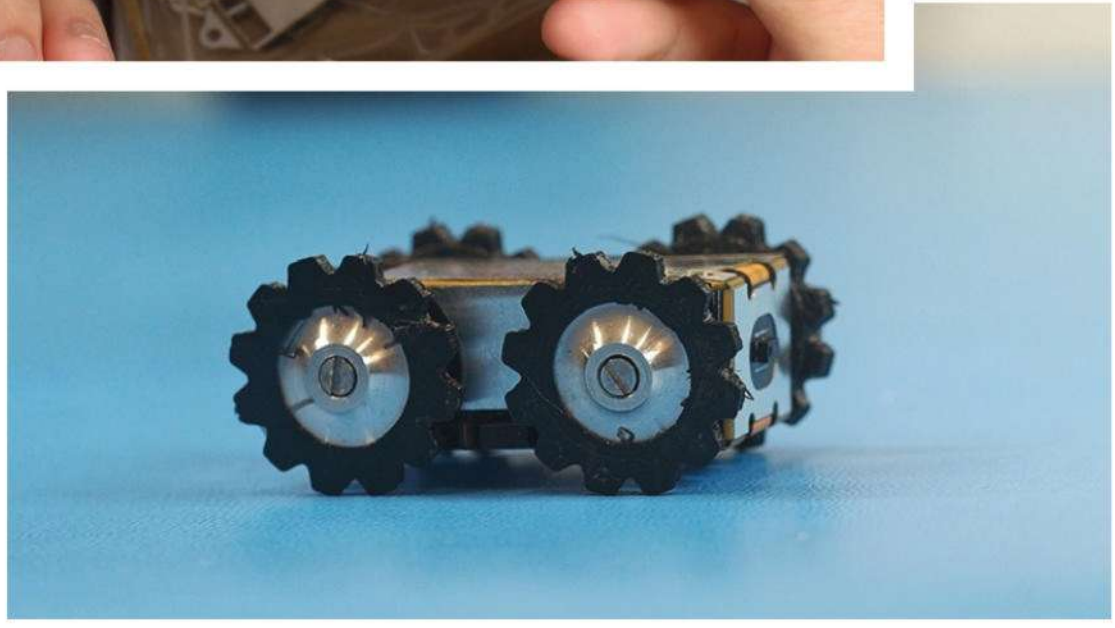
Right
This folding robot is really just a couple of lasers away from being a real-life Transformer











REGULAR

3D-printed RC car

By **Rambros** hsmag.cc/RC-car

Do you remember the Tim Burton version of *Batman*? With the Batmobile that was little more than a big jet engine on wheels? That's what this build, by Rambros, reminds us of.

It's a radio-controlled car with a 3D-printed chassis, body, steering assembly, and power transmission (including a differential for the rear wheels). Only the electronics, wheels, bearings, and the screws that hold it together aren't made of plastic.



Right
Rambros designed this awesome car in Fusion 360

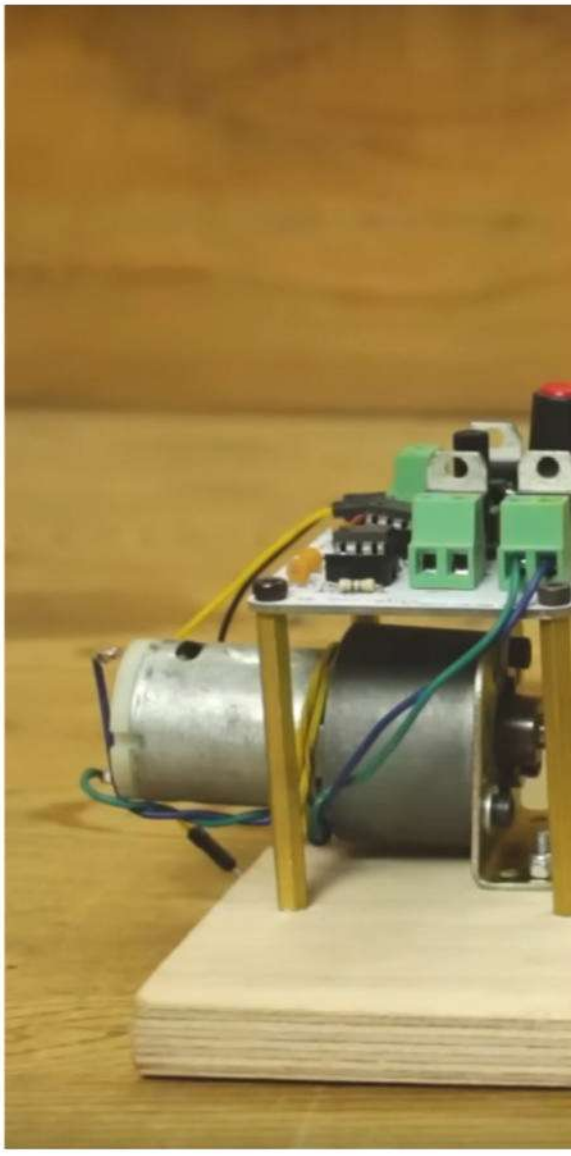


Circular knitting machine

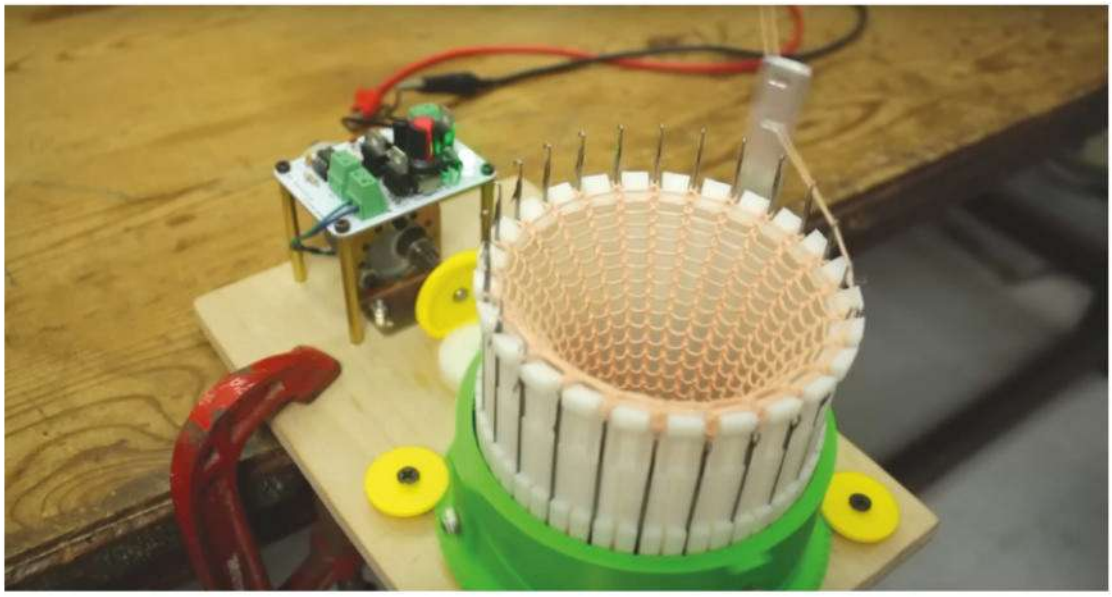
By **Mr Innovative** hsmag.cc/CircularKnittingMachine

We're always fascinated by the application of new technologies to old crafts, and this 3D-printed knitting machine is a perfect example of that. More than that, it's absolutely mesmerising to watch. Mr Innovative, its creator, designed the machine in two parts: the stator and the rotor. The rotator spins, winding the yarn around the needles, and pushing the needles up and down within the stator.

Just like knitting with needles, it's hard to get your head around, but that makes it all the more satisfying when you see it in action.



Right
Why build a fancy enclosure when you can just stick your machine on a bit of plywood?





REGULAR

Objet 3d'art

3D-printed artwork to bring more beauty into your life





This enormous object is a Stargate fourth-generation 3D printer. Its maker, aerospace company Relativity Space, claims that it's the biggest metal 3D printer in the world, which impresses us; what impresses us more is that it's currently in use manufacturing Relativity's Terran R and Terran 1 rockets, which are both almost entirely 3D-printed (including 3D-printed engines). If you were to 3D-print a rocket at home, you'd quickly run up against the height of your ceiling; to get around this, Relativity has oriented the Stargate printer so that it prints horizontally, giving it the ability to print objects up to 120 feet long and 24 feet wide.

We're used to the advantages of 3D printing in plastic, and it turns out that many of those advantages apply to printing rockets. You can iterate more quickly than with injection moulding, and use the same piece of equipment to build different parts, so you don't need a room full of kit to make several complex pieces. Relativity calls this effect the 'software-defined factory', and claims that it enables the company to build a rocket in two months, rather than the two years that it takes the rest of the industry.

Letters

ATTENTION ALL MAKERS!

If you have something you'd like to get off your chest (or even throw a word of praise in our direction), let us know at **hsmag.cc/hello**

GUITAR

The 3D-printed guitar from Prusa looks exactly like how I imagined the future would look when I was young. Unfortunately, for the most part, the future hasn't quite panned out how I'd hoped: instead of jet packs, we have social media. The world might be falling apart, but at least we can now rock out in suitably cyberpunk style while it does.

Jane
York

Ben says: We heartily agree. For the most part, the future is a bit disappointing. This guitar does indeed look fantastic, and we have a new Prusa printer that needs some testing. Let's see just how good it sounds. I just need to learn how to play the guitar.

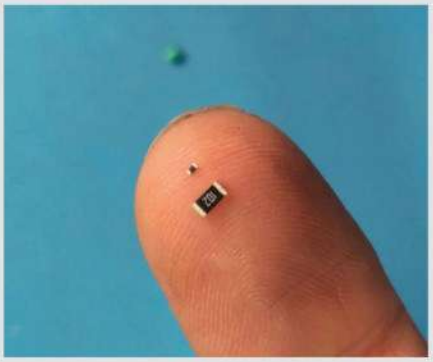


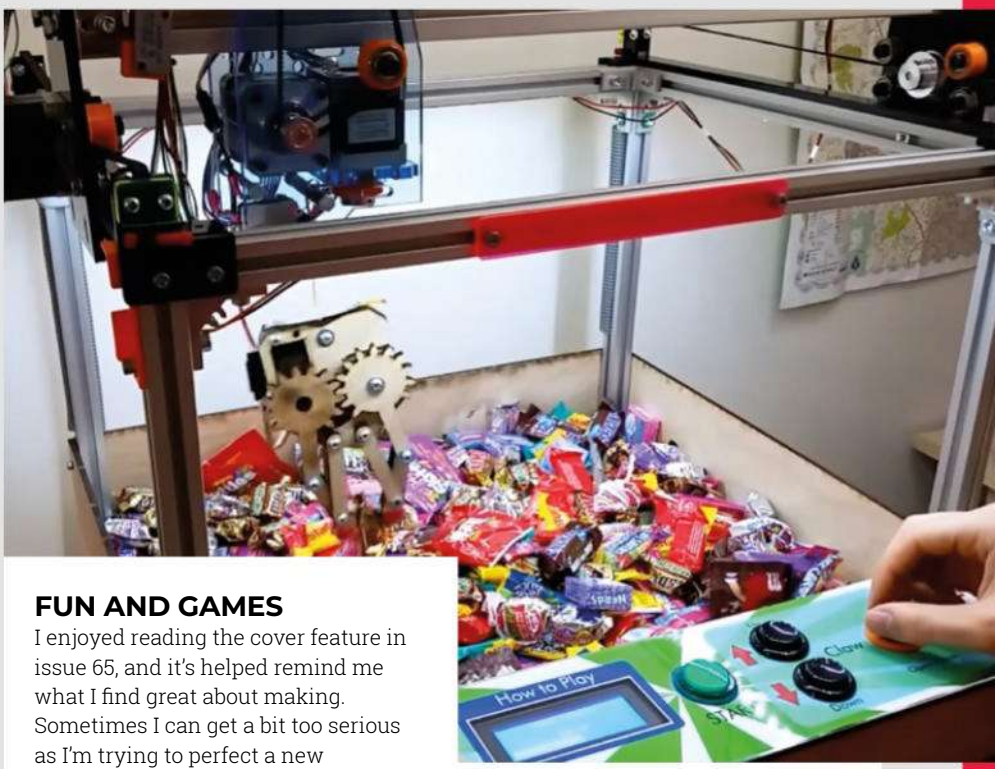
SOLDERING

I've been soldering for years, but always by poking leads through holes in the PCB. Thanks to your article on surface-mount soldering, I've decided to take the plunge and ordered a practice kit. Wish me luck!

Ali
Cardiff

Ben says: Good luck! Not that you'll need it, though, surface-mount soldering doesn't need luck, it just needs a little practice. With your kit, you'll be well on your way. It's slightly different from through-hole, so it can take a few goes to get your eye in, but once you do, you'll be fine. As more and more components only come in surface-mount varieties, soldering SMD is increasingly an essential maker skill. This issue, we're looking at going up to the next level and soldering with paste, but take it one step at a time.





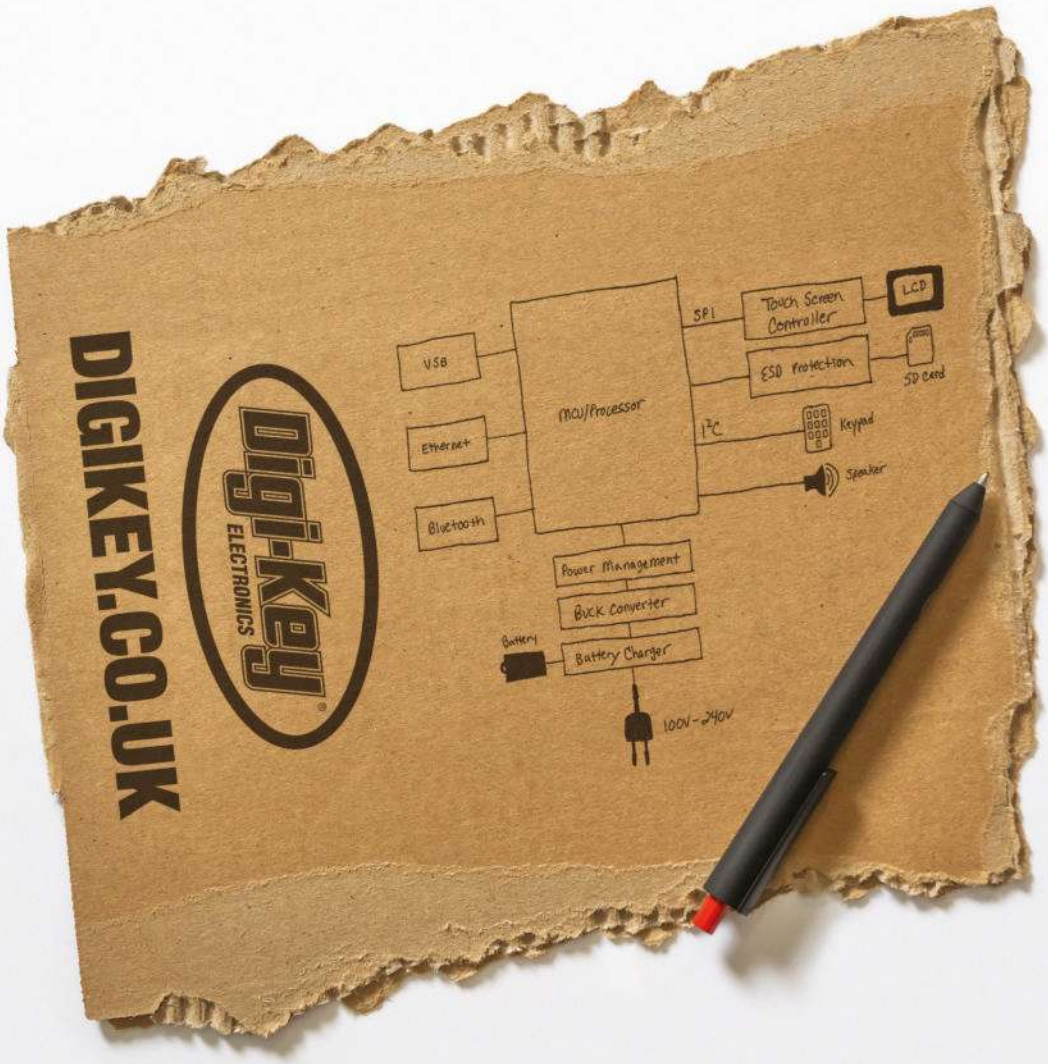
FUN AND GAMES

I enjoyed reading the cover feature in issue 65, and it's helped remind me what I find great about making. Sometimes I can get a bit too serious as I'm trying to perfect a new technique or get a finish just perfect, but actually, what I really like is just playing with technology and seeing what I can do with it. Making games is a great excuse to do this because there's no risk – if it all goes wrong, then who cares!

Ashley
Cambridge

Ben says: This is exactly what we wanted to get across in the article. Games can be an excuse to try something out or a way of showing off a particular skill. Either way, the critical thing is to have fun – both making them and playing them. As always, we'd love to hear about your builds!

IDEAS
START HERE



From millions of in-stock parts to the latest new product inventory, we've got everything you need to turn breakthrough ideas into reality.

Get inspired at **digikey.co.uk** or call 0800 587 0991.

Digi-Key ELECTRONICS

Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2023 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG 34

HOW I MADE:
CAT GPS

How one ingenious maker keeps tabs on his itinerant feline house guest

PG 40



INTERVIEW:
FRAN SCOTT

If gear held together with chewing gum is good enough for the Royal Society, it's good enough for us

PG 48

IN THE WORKSHOP

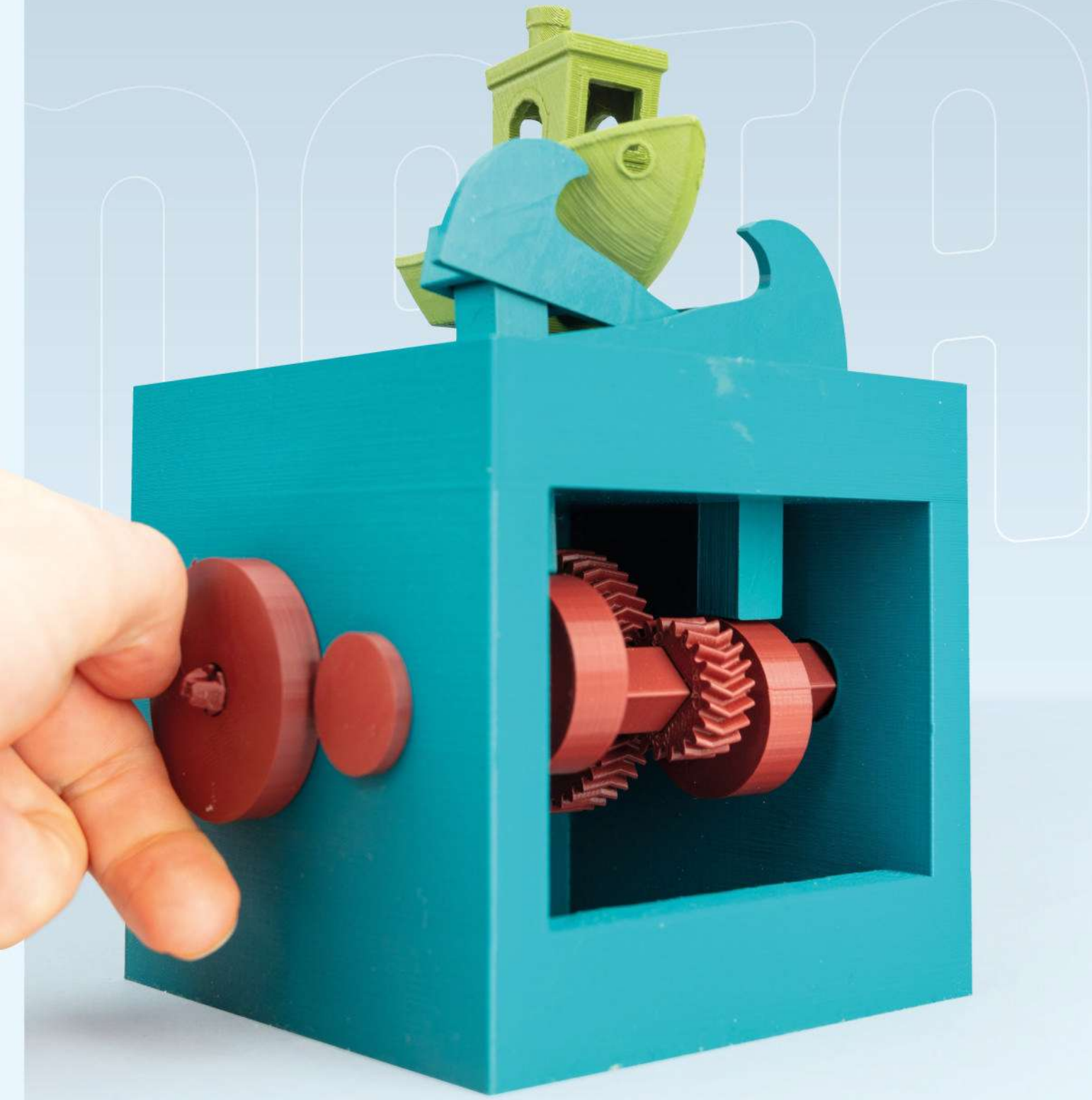
Building a battery-powered infrared camera to track night-time visitors

PG 22

AUTOMATA

Play with mechanical engineering for 3D printing





AUTOMATA

Learn mechanical engineering with a 3D printer

Generations of mechanically minded folk have played with automata. These are mechanisms that – in some way – move by themselves, and the term is usually reserved for things that aren't electronic.

They can be powered by springs or gravity, but in the case of the 3D-printed ones we're looking at, they're operated by turning a handle. The automata consist of some form of mechanism to transform this rotation into the animation for a scene. We'll see flying animals, hunting felines, and ocean waves.

These automata can show how relatively simple mechanisms can create quite lifelike movement. As well as being fun little games, they give the maker a safe space to explore different tools, techniques, and materials.

In this feature, we're looking at 3D-printed automata. It's common to think of 3D printers as simple replicators that don't require any thought or effort in their use, but this isn't the case. Like all tools, 3D printers require skill and knowledge to let the user get the most out of them, and obviously, 3D design is a whole set of skills in its own right. Automata give us a space for experimenting with these where there's not much risk if things go wrong. Getting your automata to work will require thinking about forces (and the orientation of these forces), tolerances, and the mechanical properties of the materials we're working with – as well as the mechanics of the design.

We're going to take to the skies, catch mice, sail the seas and more with just the help of some filament and a printer. Let's see what it takes to make your prints move. →

PICKING PLASTIC

We were able to print all the parts for these automata in PLA, but this isn't necessarily the best plastic for the job. Nylon's low friction or the toughness of PETG or ASA might make them a better choice for some of the parts.

One significant advantage of PLA is that it comes in a wide range of colours, so you've got plenty of stylistic choices. That said, if you don't want to clog up your cupboards with spools of different colours, you can paint your parts. This probably won't work well for moving parts such as gears, but for some other bits, it's a quick and cheap way to add more colour. The cat and bat models in this article are painted as we didn't have a suitable colour PLA to hand.

EAGLE IN FLIGHT

Gears, linkages, and a realistic flight

hsmag.cc/flyingeagle

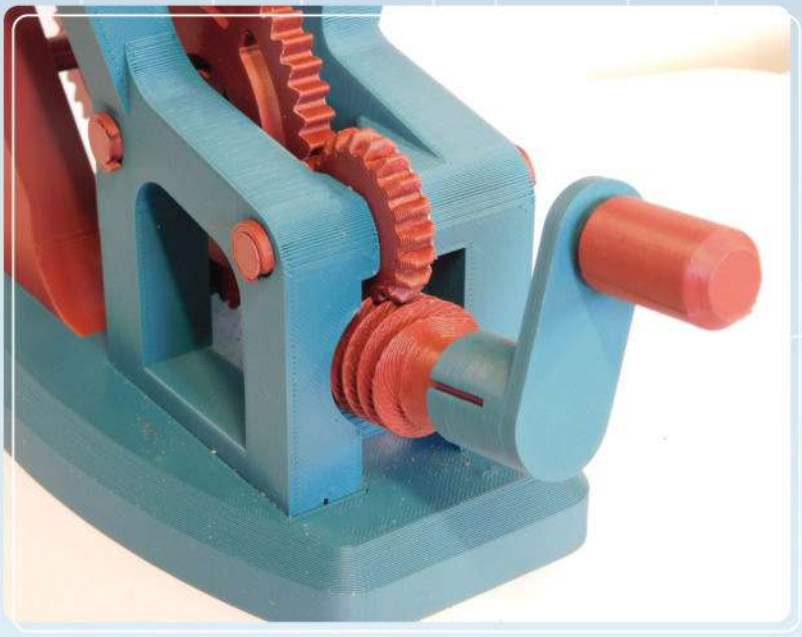


This machine is frankly a ridiculously complex mechanism for a 3D print. There are detailed instructions on the printables site at **hsmag.cc/flyingeagle**, but friction is the main enemy on a build like this. You need to make sure that your parts fit together smoothly. A big part of this is the point where your print lies on the print bed – you can get a bulge on the bottom layer known as elephant's foot.

If you haven't got your bed perfectly level with the right Z offset, you can end up with the first layer spreading out a little more than the other layers. When parts have to fit together snugly, this can cause problems. There are two ways around this: tune your printer, or rectify this problem afterwards with a knife or sandpaper. You might need to do a bit of both.



Above
The linkage is a bit stiff in practice, so you'll need some sandpaper and patience to get it flapping



Above
The large gear reduction means you have to turn the handle a lot for the eagle to fly

Even with a perfectly tuned printer, you might find that you need a light sanding, particularly on the peg that goes through the eagle's body.

The friction on this eagle puts a huge stress on the axles. We found that we had to print these at 100% infill to make them strong enough.

DRIVE CHAIN

There are a lot of gears in this build, but that's really the most straightforward part of it. The handle turns a worm-gear that drives a gear chain that ends in two bevel-gears taking the rotation into the wing mechanism. This drive mechanism does two things: it takes the rotation from the handle and moves it to the place it's needed, and it reduces the speed from a fast handle wind to a slow wing-flap.

The movement of the wings looks like one big linkage of levers, but it's probably best thought of as two mechanisms. The bird with wings, and the levers that drive this.

Linkages of stiff rods can create a range of movement. However, they can be a little difficult to design because it can be hard to work out exactly what arrangement of rods creates what movement.

The specific movement this eagle uses is known as a Chebyshev lambda linkage. This was originally designed as a linkage to turn rotational movement into a straight line. Well, almost a straight line, it's more like a flattened D shape. In this model, this movement is where the lower lever joins the wing. This moves the bird up and down and the slight side-to-side movement is used to extend the wings for the downstroke and extend them for the upstroke.

Once you've got everything up and running, it's an absolutely delightful automaton. It's almost impossible not to turn the handle each time you pass it.



Left
The bird stretches out its wings for the downstroke

FLYING BAT

A simpler flying machine

hsmag.cc/flyingbat

This bat automaton used a pair of crank-shafts to both raise and articulate the wings. It's almost all 3D-printed, but you will also need some 3mm wire (which makes the pivots) and four M3 screws (about 10mm or so). We did experiment with replacing the 3mm wire with 1.75mm filament, and it just about worked but was a little janky – we suspect it wouldn't last long. If you happen to have some 2.85mm filament tucked away somewhere, it would probably do the job perfectly.

The two cranks are kept in sync by gears on the back. Each crank-shaft drives two rods. One (which has a short joint on the end) plugs into an appropriately shaped hole in the wing. The second joins an L-shaped linkage on the main pivot and drives the opposite wing-tip.

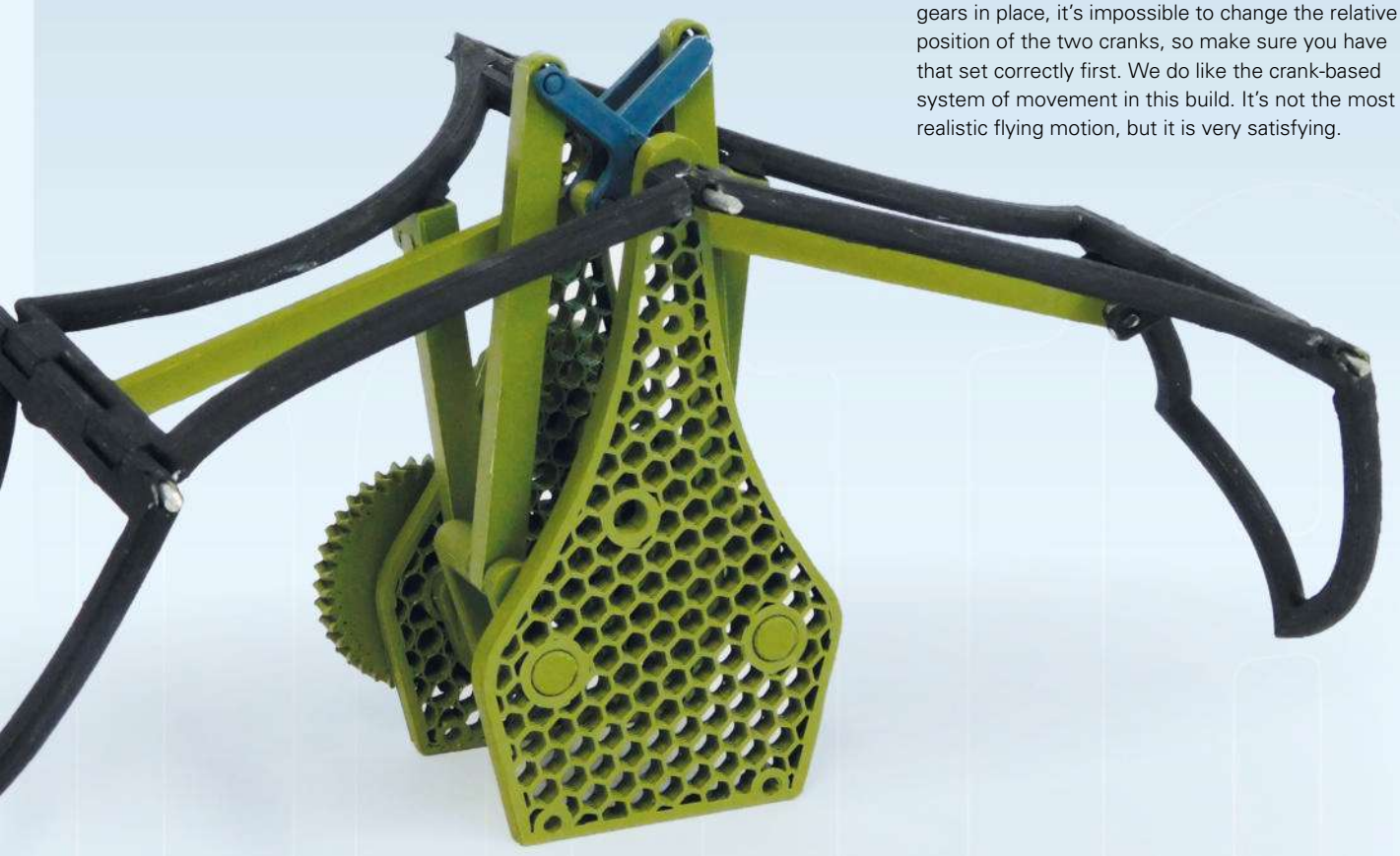
There are very few instructions on the project, and the photos don't do a great job of showing you how to make it. However, we've done the hard work of figuring it out.

The one instruction that Printables user Kingfisher does give us is a good one. The front and back panels get their distinctive hexagon look by printing them with no top and bottom layers, and just hex infill. The pattern isn't in the model, it's exposed infill.

There's not a lot to the build once you've worked out how the mechanics go, which you should see from the images here. However, you do need to glue it together, so once you've put it together, there's no going back if you've made a mistake.

We've taken photos of the key parts which should help you put things in the right place. The only particular trap we found is that once you've glued the gears in place, it's impossible to change the relative position of the two cranks, so make sure you have that set correctly first. We do like the crank-based system of movement in this build. It's not the most realistic flying motion, but it is very satisfying.

Below
The linkage looks a bit like a real bat's shoulder muscles





CAT & MOUSE

Predator and prey

hsmag.cc/catandmouse

This model was uploaded seven years ago by Thingiverse user MakerMatic. It's designed to be push-fit together, but we found the joints very loose, so you will also need some glue. A handle turns a camshaft, which has two cams on it. Both pushing up levers – one leaver is the cat's arm, and the other is its tail. At the other end of the shaft is a bevel-gear that rotates a platform with a mouse on it.

Looking at the creator's video, it appears the gears should fit together properly, but due to a happy accident, they don't on our build. This means that rather than moving around smoothly, the mouse does little darting runs as the gears catch then skip.

This model is easy to print and make, but there is a slight problem – one that you'll find in a lot of 3D-printed mechanisms – Z axis strength.

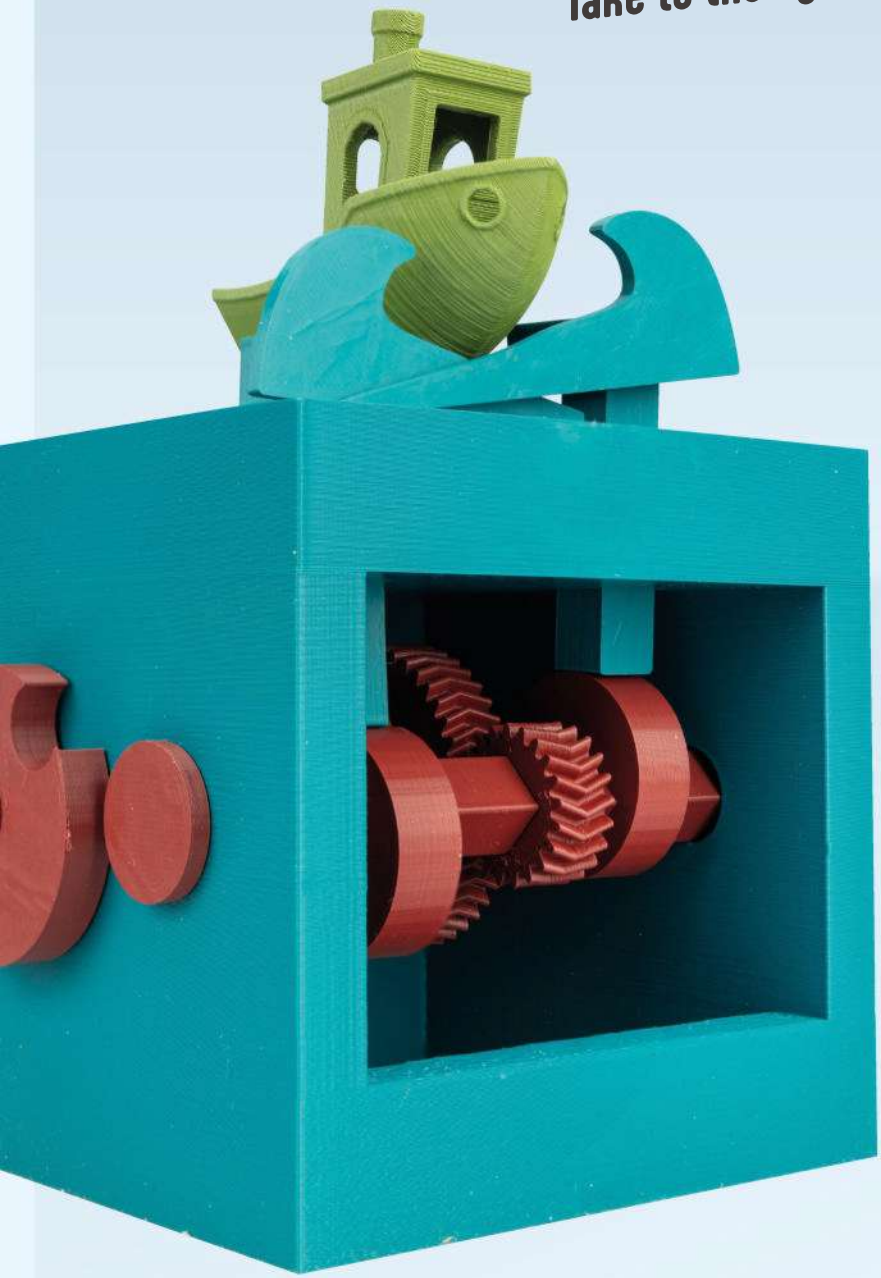
3D-printed parts can be strong in some directions, but they're weakest at the layer lines. Thin parts are particularly weak. In this build, there are a few parts that most naturally print this way – pins to hold the levers in place and the handle. There's no good way around it. Some makers use non-3D-printed parts for this (such as wire or bolts), but that can make it harder to replicate the build.

Compared to the other automata we've looked at here, the movement is much faster. Although it's not truly erratic, the flick of the cat's tail does look like a real hunting feline.

Above
Turn the handle to see the cat try and catch the mouse

BUILD YOUR OWN SAILING BENCHY

Take to the lightly-animated seas



As a magazine about making things, 3D printers pose a bit of a conundrum. After all, all you need to do is load up the filament, slice the model, and press print. However, that's not really the full story. There's also the mechanics of how it works, and you can look at how to design and tweak the model. In this feature, we're going to cover all these angles, and we'll start with printing and assembling the models.

You can download all the bits you need from Printables at **hsmag.cc/automaton**.

You'll need one of everything except:

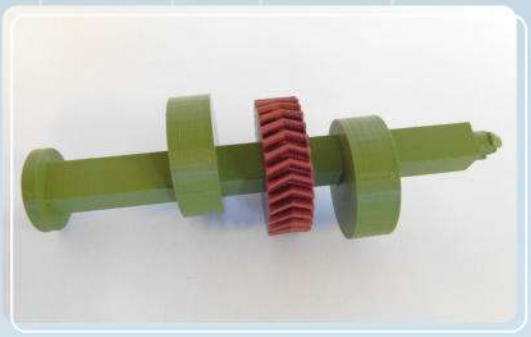
- 3 × cams
- 3 × cam followers
- 2 × axles
- 2 × axle handles

We've taken the slightly odd decision to use square axles. Squares are not generally known for their ability to rotate smoothly in a round hole. While there's a bit of clunking, they do actually rotate reasonably well. The square axle does mean that the cams and gears can slot on without the risk of slipping. If you're assembling it permanently, you might want to add a drop of superglue to hold them in place, but it runs fine without this and that allows it to be completely disassembled to change the gears of cams to change the movement (which we'll look at later).

First, push the rear axle through the hole and thread on a cam the larger gear – it doesn't matter which order they go on in.

You can now push on one of the axle handles and it should clip into place.

You can now push the second axle through the hole. On this you'll need to put on a cam, then the smaller gear, then a third cam. It's important at this point to get things the right way up, as you can get different effects by having the cams the same way up



Above
The axle and handle snap together

(having both waves rise together) to having them at different angles. We like the effect of having them 90 degrees out.

You also have to be careful how you put the gear on for two reasons. Firstly, the herringbone has to be the right way around to mesh with the other gear, so you might need to flip it over. Secondly, you won't be able to mesh the gears once the axle is in place, so you'll need to mesh the gears first, then feed the axle through.

With those in place, you can clip the handle on the end of the axle and everything should be linked together. Turn one of the handles (either one) and it should all just work.

You now just need to add the bits that move. To do this, position the cams below the holes, drop the followers through the holes, and put the waves and Benchy on top of the followers.

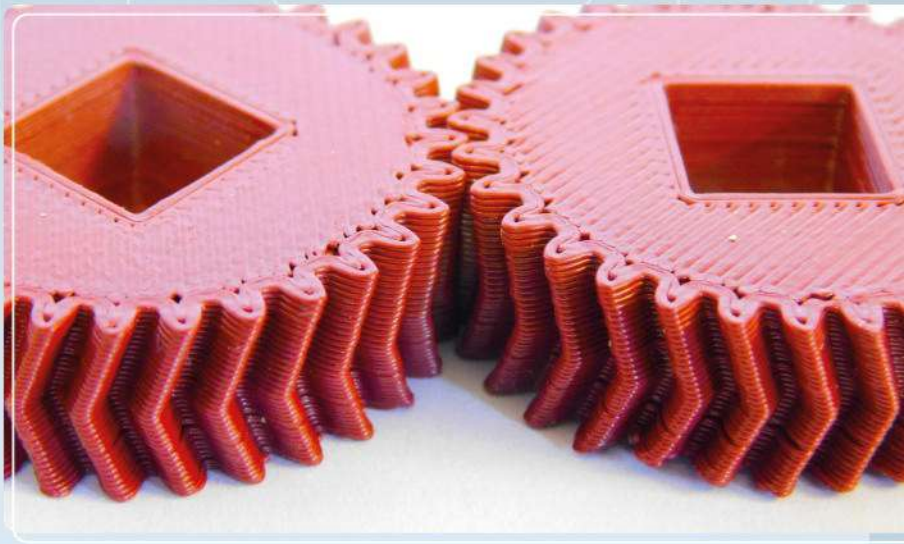
At this point, you should be able to put your finger in the hole on the wheel and give it a spin. Benchy should go up and down, and the waves should break over the bow.

That's how you make it – let's now take a look at how it works.

It's not complicated. There are basically two different types of mechanisms in the box: gears and cams. The gears connect the two axles, and the ratio of teeth on the gears determine the ratio of the speed of the axles. In our case, one gear has twice as many teeth as the other axle, so that axle spins at half the speed of the other axle.

Cams are a way of converting rotational movement into linear movement. The ones in our design are off-centre circles, but they can be many other shapes. The important bit about a cam is that as it rotates around the axis, the surface is different distances from the axle. This makes the cam follower move up and down. They can have more than one protrusion that makes them go up and down multiple times in a single rotation. The followers sit on top of the cams and transfer the linear movement to the object on top. In our case, the followers are held down by gravity, but they can also be held down by springs.

The Benchy itself is also modified slightly and rocks back and forwards on a pivot.



Above
The square axles makes it easy to keep the cams and gears secure

DESIGN

We're not going to go through the whole of the design of this automaton. If you're interested in becoming more familiar with the process of 3D design, our free e-book, *FreeCAD for Makers* (**hsmag.cc/freecadbook**), has details about how to go about this.

However, we will look at the two key parts of the mechanisms and how to modify them.

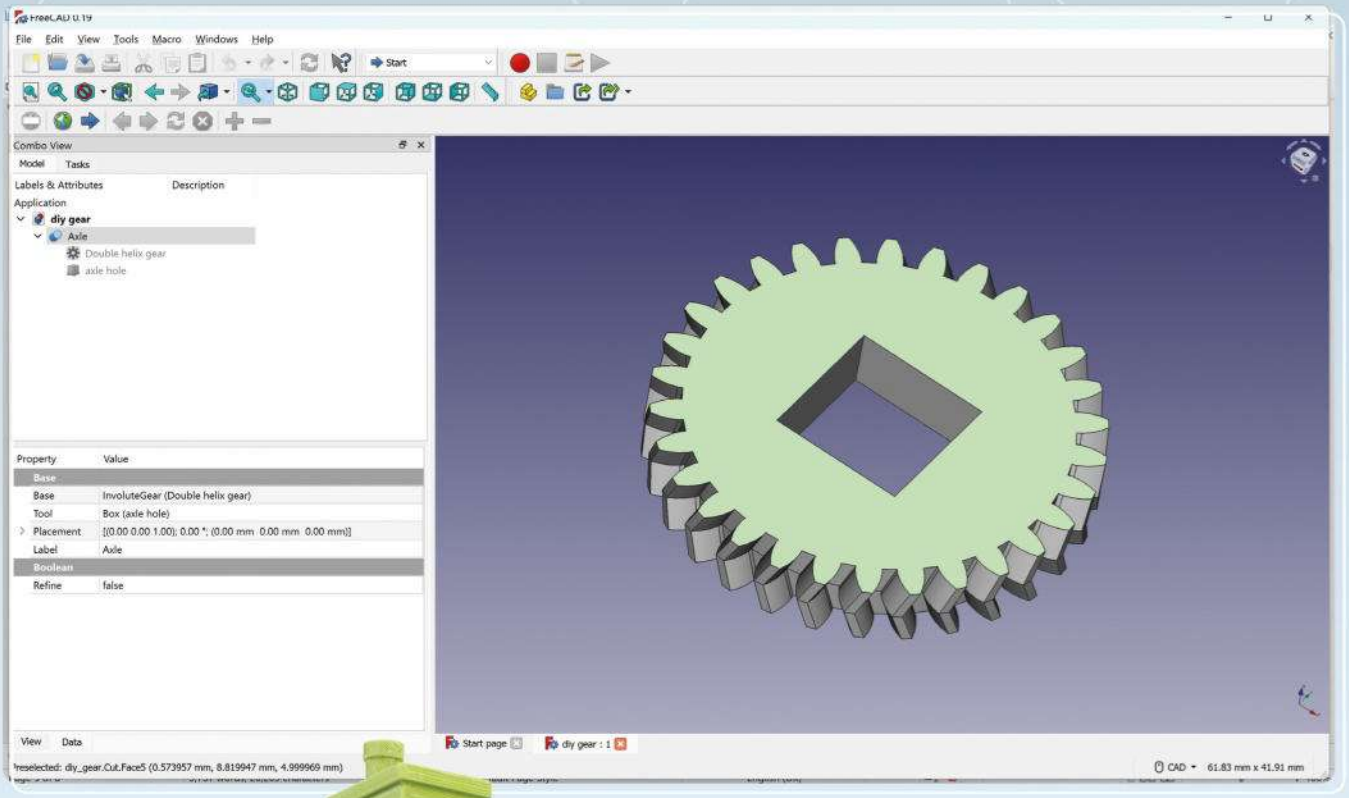
The cams are the easiest thing to adjust. The axles are 9.9 × 9.9mm. This means that you need a square hole slightly larger than 9.9 × 9.9mm in your cams. How much larger depends a lot on your printer. We found that a 10.2 × 10.2mm hole gave an easy fit even if there's a bit of elephant's foot, but you might be able to get away with a little smaller. It doesn't have to be particularly tight to stay in place.

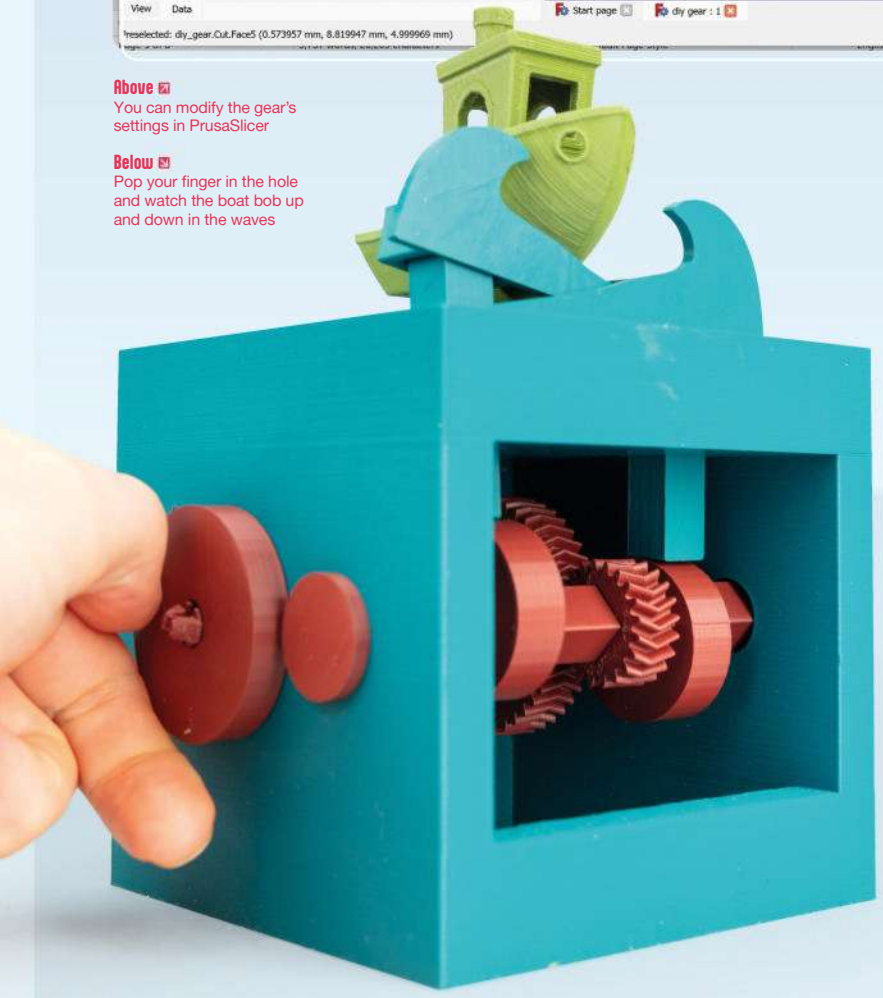
You don't even need to venture into CAD software if you don't want to. Let's take a look at how to create an off-centre square cam using PrusaSlicer. This cam will give your boat a much bumpier ride than the circular cams that the original design comes with, but all boats experience storms sometimes!

Open PrusaSlicer, and on the blank print bed, right-click then select Add Shape > Box. This will add a 1-inch cube to the print bed. In the left-hand pane, you can change the size, but before you do, you'll →

> "BENCHY SHOULD GO UP AND DOWN"

FEATURE





Above
You can modify the gear's settings in PrusaSlicer

Below
Pop your finger in the hole and watch the boat bob up and down in the waves

need to click on the padlock otherwise it'll scale it proportionally in every dimension.

Enter a Z height of 10 mm, and X and Y as 28 mm.

We now need to add the axle hole. Right-click on the box and select Add Negative Volume > Box. This now creates a shape that will be subtracted from the other shape when it's sliced. Again, you can use the left-hand pane to resize it and then move it into position. Where should it go? That's up to you. Near the centre would give four bumps per rotation, whereas near one of the edges would give a rise and fall with four bumps along the way. What effect do you want to create?

GRINDING GEARS

The Benchy is controlled by one of the axles, and the waves by the other. We can control the relative speed of the two by changing the gear ratio. This means you can have the waves go faster, slower, or the same speed as the boat. What effect do you want?

In order to create gears at a different ratio, we need to dive into FreeCAD. We're using FreeCAD 1.9 with the Gears workbench. We'll guide you along here, but if things seem strange and unfamiliar, it might be worth reading through the first couple of articles in the *FreeCAD for Makers* book to familiarise yourself with the tool.

We first need to install the Gears workbench. You can do this by going to Tools > Addon Manager, and scrolling down to freecad.gears. Highlight this and click Install. Once installed, you can use the

Workbench drop-down (which will probably show the Start workbench) to select the Gear workbench.

Now you can create a new project (File > New) and we can start building our gears.

In FreeCAD terms, the gear we're creating is an external involute gear. You can create one of these by clicking on the left-most yellow icon in the Gear workbench.

Now we need to make sure that our two gears fit in the space we have. The distance between the centres of the two axles is 72 mm. However, you can't just create a gear with an arbitrary radius, because it might not have a whole number of teeth. Instead, there are two parameters which determine a gear's size: the module and the number of teeth.

The module is the size of the tooth, and both gears must have the same size module in order to fit together. The pitch diameter of a gear (which excludes the bit that overlaps with the gear it meshes with) can be calculated as the module multiplied by the number of teeth. The maths is obviously simpler if we use a module of 1. Since we're only interested in the radius, but the radius of both gears, the whole formula collapses and we just need to make sure that the sum of the teeth on both gears is the distance between the centre of the two gears – in this case, 72.

GOING FISHING

One slightly unusual feature of the gear is the 'herringbone' arrangement of the teeth. We'll be honest and say that the appearance was a significant factor in choosing this style of gear, however, it does have a practical use – it locks the gears together, meaning they won't slide sideways out of alignment, even if they're not fixed to the axles. Of course, a drop of glue would do the same, but as we said, we like the way this looks.

You can create a herringbone gear in FreeCAD by selecting the gear and scrolling down through the properties until you get to the Helical section. The Beta parameter is the angle of the teeth (we went with 20 degrees), and set Double Helix to True to enable the teeth to angle in both directions.

This will create a gear but with no hole for an axle, so again, we need to add a square hole in the middle.

Now use the Workbench drop-down to change from Gear to Part.

In FreeCAD, there isn't quite the same concept of negative volume that there is in PrusaSlicer (at least not in the Part workbench). Instead, we can create a regular box and use a cut to remove the box from the gear. First, you need to switch to the Part workbench, then click on the yellow cube to create a cube solid.

You can set the length and the width of the cube to be 10.2 mm. However, the cube will be positioned with one of its corners at the origin while the gear will be positioned with the centre of the gear at the origin. This isn't what we want. You can simply calculate the offset for the cube (-5.1) to put it in the right position; however, you'll have to adjust this if you want to change the tolerances. Another option is to use the expressions.

In the Base section of the cube properties, click on Placement to open up the options for X, Y, and Z. Click into X, then press the 'fx' button to open the expressions box. In here, you can type:

```
-Width/2
```

And in the Y box, you can enter:

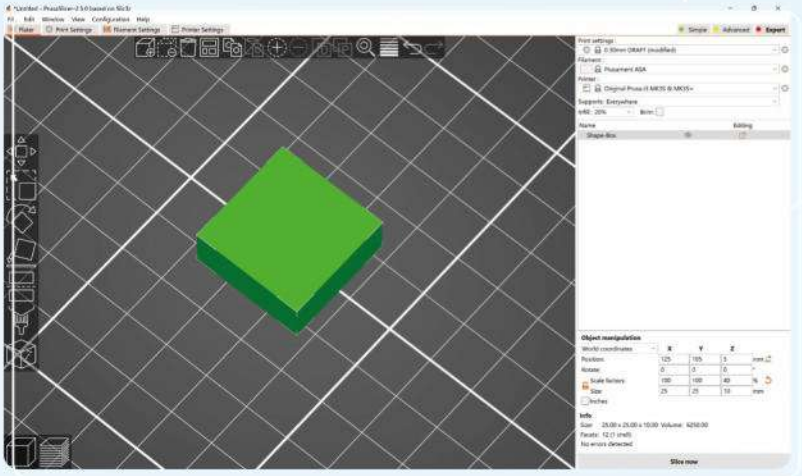
```
-Length/2
```

The cube should now be centred on the gear. If it isn't, right-click on the cube in the cube in the left-hand pane and select Recompute, as it sometimes doesn't do it instantly.

We've uploaded a file with this all setup as **diy gear.fcstd** to the Printables project. You can download this, and then just modify the number of teeth. To do this, open the file in FreeCAD, then open the DIY Gear > Axle drop-down and select Double Helix Gear. This will open the properties box in the left-hand pane, and you can update the number of teeth.

Have a play with this automaton and see the effects, then set forth and design your own. Explore your own mechanisms using gears, cams, cranks or whatever you like. You'll have some fun, and hopefully learn a bit along the way as well.

> "WE LIKE THE WAY THIS LOOKS"



Below You can design really simple parts directly in PrusaSlicer

SUBSCRIBE TODAY

FOR JUST £10



Get three issues plus a
FREE Raspberry Pi Pico W
delivered to your door
(UK only)

hsmag.cc/FreePico

Subscription will continue quarterly unless cancelled

NEW INTERNATIONAL OFFERS!



Get 6 issues plus a **FREE Raspberry Pi Pico W** for just...

£35 USA and EU

£40 Rest of the world

Head to **hsmag.cc/subscribe** to get yours today!



FEATURE

HOW I MADE

By Sahas Chitlange

PET TRACKER

Keep tabs on a wayward moggie

Pets have an incredible ability to become an integral part of the family in no time. We love our furry companions unconditionally, and are willing to go to great lengths for them. However, these adorable little mischief-makers can sometimes be a handful. One such example is the cat, a four-legged mischievous imp that takes pleasure in doing things it's not supposed to.

I happen to have the mischievous orange variety of these furry companions. My little troublemaker goes by the name of Pumpkin, and despite (or possibly because of) his shenanigans, I love him dearly. There have been instances in the past two years where Pumpkin has gone missing, causing me to panic. It has made me realise that it would be useful to have a tracking system in place to locate him if he ever takes off again.

I looked for commercial solutions and wasn't too impressed with what was available. Most solutions fell into the following categories:



Left
The located cat

1. They used non-GPS-based systems for localisation. Think about all the 'works with iPhone'-branded trackers that you see at the pet stores. These require an iPhone to be in close proximity to the pet to track its location.

2. The second group consisted of GPS-based systems, but they had low refresh rates, which meant that the tracker's location was only updated and sent back to you a limited number of times. While they were still somewhat useful, they were not ideal.

3. The third category consisted of misleading advertisements for devices that were marketed as 'GPS' trackers, but actually relied on Bluetooth or other small-range radio signals to estimate the distance to the pet. Unfortunately, these types of devices are pretty much useless, since location tracking based on received signal strength indicator (RSSI) is not accurate and can be easily affected by interference from everyday objects made of metal and concrete.

The commercial options didn't really make the cut for me. This is how I began my research into asset-tracking systems. My goal was simple: to build a device that would be on Pumpkin, connected to the internet, and would send me a live location fix any time I requested it.

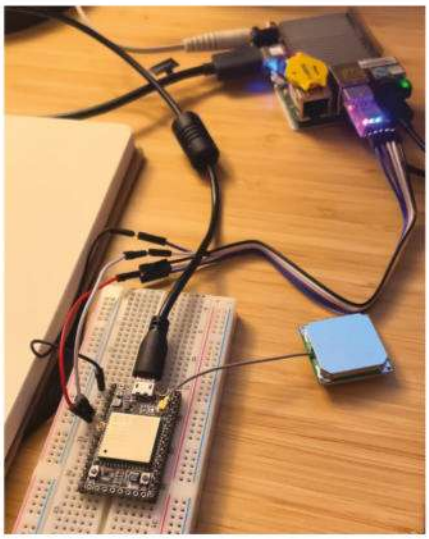
HOW?

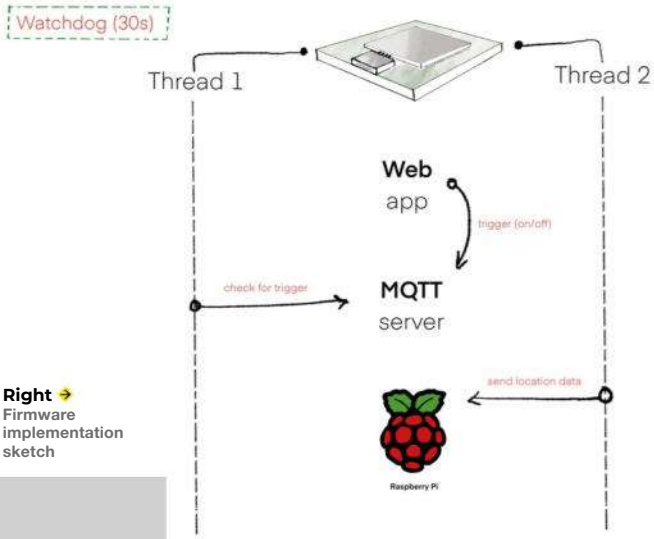
The first challenge was to choose a GPS chipset that came with cellular connectivity. Having previously built a GPS shoe (**hsmag.cc/GPSShoe**), I had some familiarity with the SIM808 module, but I found that it was too large for my current application and required an external microcontroller to run the firmware, adding extra circuitry and reducing its compactness.

I did some research on available modules, and it seemed like the A9G chip was a good choice (spoiler, I was wrong). It promised GPS, GSM, and a CPU to run the firmware and it was pretty compact!

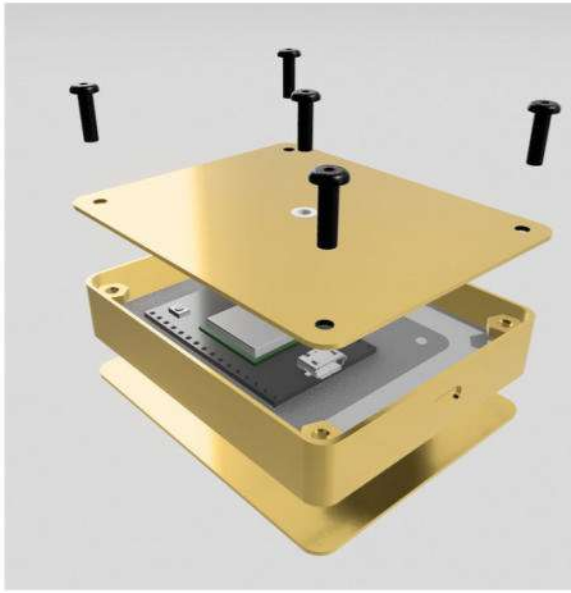
Insights from the future: the A9G module had three big disadvantages. One, the manufacturer has discontinued support for it and the documentation is very lacking. The

Below
Uploading MicroPython firmware to the A9G





Right
Firmware implementation sketch



Above
CAD design of the enclosure

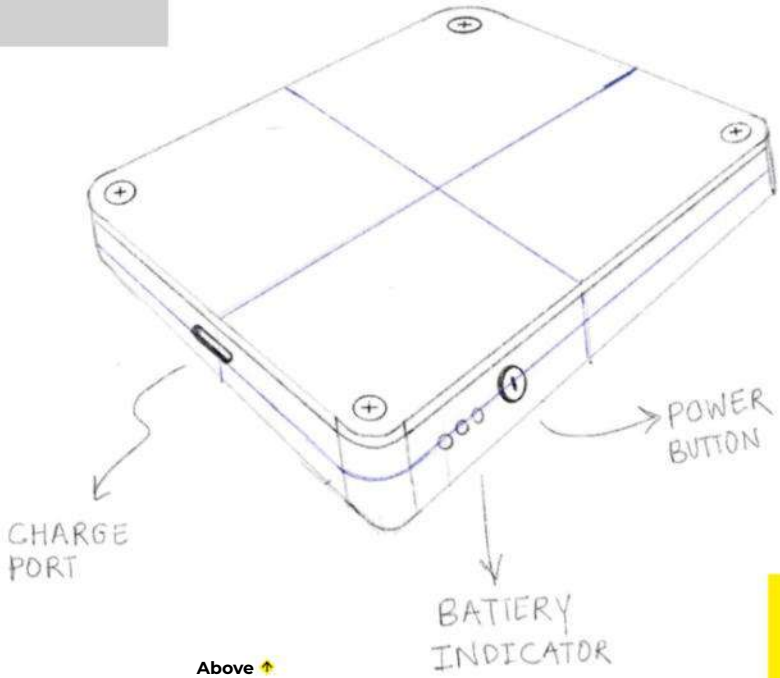
chip only supports 2G (discontinued by most telecoms carriers in the US), and finally, power consumption is a bottleneck because the module wasn't designed for low-power use. Low-power LTE-M modules are a much better path to pursue.

After finalising the GPS chipset for my project, I initially attempted to develop firmware using the C SDK (software development kit) that A9G manufacturers ship. However the 'time to Hello World' with the SDK was longer than I had hoped due to lack of documentation and so, to accelerate the development process, I decided to switch to MicroPython for A9G by Pulkin (**hsmag.cc/MicroPythonA9G**).

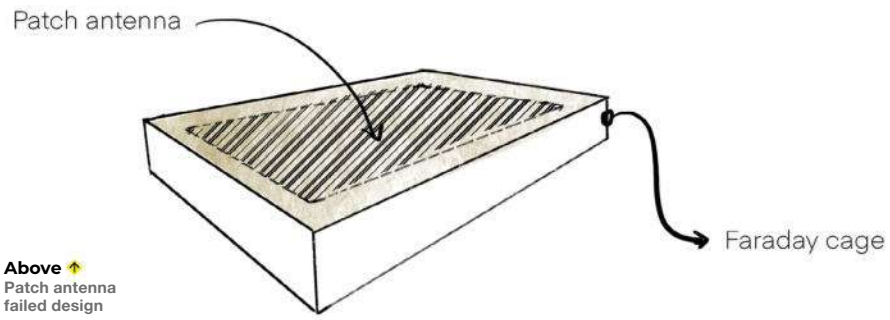
After switching to MicroPython and successfully running a 'Hello World' program on the A9G chip, I began brainstorming ideas for my firmware algorithm. I aimed to create a firmware that would allow the device to sleep most of the time and listen for messages in a separate thread. Upon receiving the appropriate message, I wanted the device to initiate the tracking process.

I chose MQTT (Message Queuing Telemetry Transport) as the communication protocol between the MQTT broker and my tracking device. It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth. It is a very popular protocol, and some experts would call it the standard for IoT (Internet of Things).

MQTT is amazing without a doubt, but most MQTT implementations are based on



Above
Sketch of GPS enclosure



Above
Patch antenna failed design

TCP (Transmission Control Protocol) and per my current research, TCP is not very promising for very low-power devices as it is a 'connection' based protocol. It means that there is a lot of overhead just trying to keep the connection alive. This takes a huge toll on battery consumption, hence lowering the battery life. I'm still researching more protocols for optimising low power, and the UDP-based CoAP / LWM2M certainly look more promising.

I designed my firmware with two threads running in parallel. Both threads sleep for most of the time and periodically wake up to check for incoming data. When an incoming message is detected, and if it is a command to initiate the tracking process, the GPS engine is started. The device then waits for a GPS fix and begins sending location updates to the Traccar GPS server (**traccar.org**).

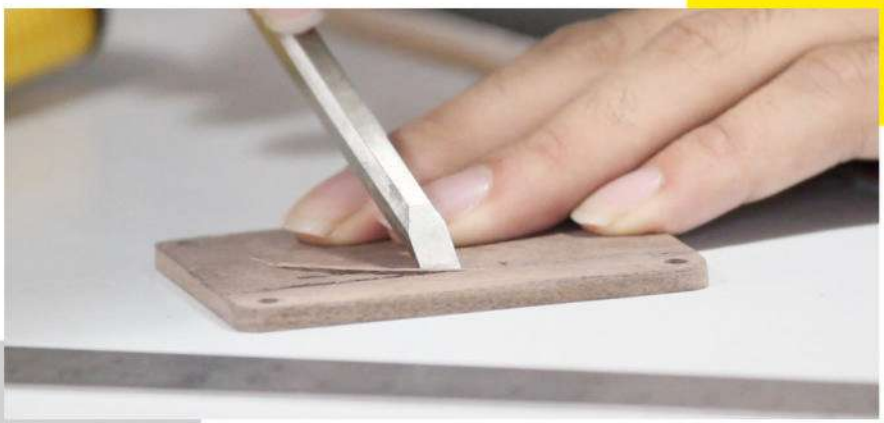
I chose the Traccar GPS server to collect and visualise the GPS data, and it proved to be an excellent choice. Traccar supports multiple protocols and is open-source, making it a flexible and customisable option. Furthermore, it can run on a Raspberry Pi, which means that I could host it myself, significantly reducing the running costs of the tracking system. To ensure constant accessibility of my Raspberry Pi from the internet, I had to perform some dynamic IP wizardry.



Above
Revised double dipole antennas inlayed into wood

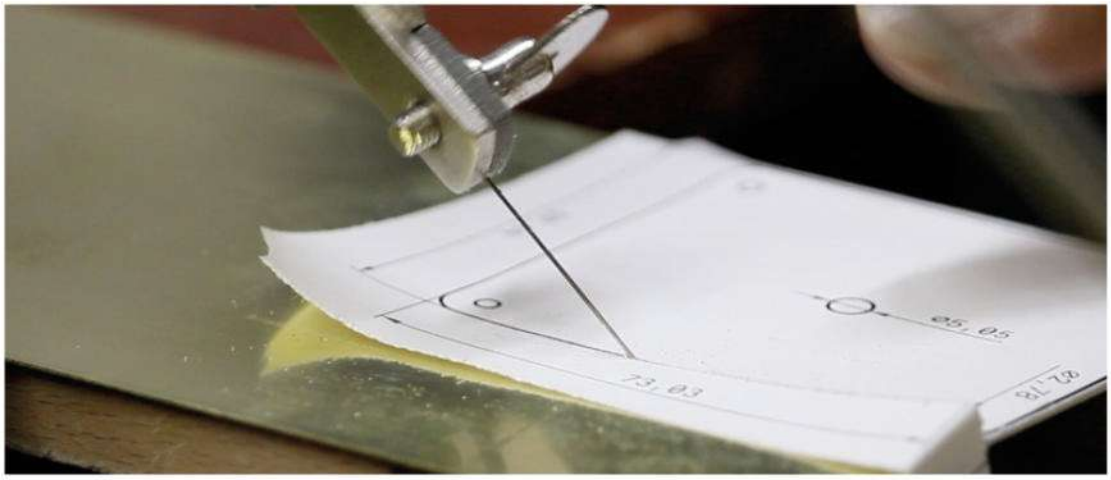
After successfully running my GPS server on the Raspberry Pi, I needed to make it accessible to the internet. This task proved to be complex and time-consuming, but eventually I found a solution by utilising ngrok. I exposed the two ports essential for the GPS tracker connection and the user interface (UI).

There were two reasons why I chose to host my own server. Firstly, it presented an excellent opportunity for me to learn about networking, especially as exposing a localhost application to the internet can be challenging, particularly for someone



Right
Making an inlay in walnut

FEATURE



Right
Cutting the brass sheet with a jewellery saw

who is not familiar with networking techniques. Secondly, I wanted people to be able to replicate this project at a low cost – and hosting a server, while relatively inexpensive, may not be the preferred option for some. However, I must note that if you intend to use this system for tracking purposes and require reliable performance, hosting it on a cloud service is a better choice.

Once the software portion was working end to end, it was time to build the hardware. I started off with a pen-paper sketch of the enclosure design. Drawing inspiration from my design mentor Dean Bacalzo, I learned to start with rough sketches and keep an open mind throughout the design process.

"MAINTAINING THE BRASS AESTHETICS WAS IMPORTANT"

In my opinion, product design is an essential component of any project, even if it is a DIY. Without it, a build is incomplete. I dedicate a decent amount of time refining the form, and carefully selecting the materials used for my projects.

My initial plan was to utilise a GSM dipole PCB antenna that came with the A9G chip for the cellular radio and create a customised patch antenna for the GPS radio by using the top plate as the ground plane. However, I overlooked a critical flaw in this design: the brass enclosure essentially being a Faraday cage would keep the radio signals from reaching the antenna, rendering it useless.

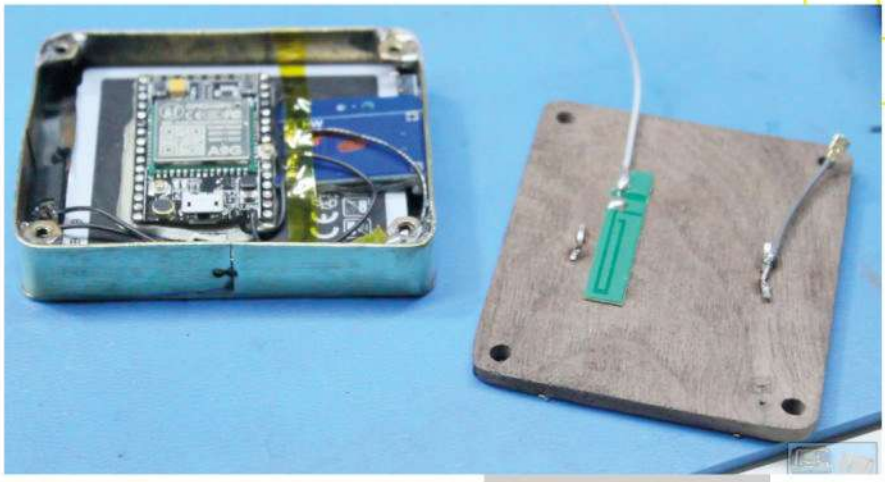
Maintaining the brass aesthetics was important to me, but I also recognised



Above
Soldering the enclosure together



Right
Adding the nuts to the enclosure



Left
Putting it together

that having a good-looking box would be pointless if it didn't function properly. Therefore, to balance aesthetics with engineering constraints, I decided to construct the top plate out of walnut wood (as wood lets RF signals pass through) and inlay two custom dipole antennas that I built myself. Although I felt a bit foolish initially for making an error in my design, I was proud of how I was able to iterate and devise a solution that worked.

I understand that some people may have questions about why I decided to construct the entire enclosure out of brass, even though it almost led to design failure. I have given this some thought and have come up with two reasons. Firstly, I grew up in a small city in India where I did not have access to advanced tools or technology. This experience has encouraged me to limit my dependence on high-tech tools, and to only use them when they are absolutely necessary. Additionally, I believe that there is a unique quality to handmade objects that cannot be replicated by machines.

WHAT'S NEXT?

This project led me down a rabbit hole of learning about asset trackers and low-power IoT, and although I am proud of what I have created, I believe there is still ample room for improvement. Battery life and physical size are the two main areas I am focused on enhancing. I have been researching low-power cellular design and exploring how to leverage the new LTE-M technology. Currently, I am prototyping Version 2 (V2), and I am already seeing promising results with the low-power nRF9160 chip. Even in its untuned, non-optimised state, my prototype only consumes 400 µA (micro-amps). By simply replacing my current A9G chip with the nRF9160, I can extend the battery life from twelve days to over nine months. Additionally, I plan to significantly reduce the device's size.

Over the next few months, I will be delving deeply into the field of low-power design for IoT. This project has ignited my passion for exploring IoT with new communication protocols such as CoAP / LwM2M, and I am thrilled about the possibilities. In fact, I am even contemplating pursuing a PhD in this area, as designing wearables and IoT devices has always been an area of interest for me. I cannot wait to see what the future has in store!



Left
Assembled

HackSpace magazine meets...

Fran Scott

You might recognise Fran Scott. She's an engineering expert who pops up on *Lego Masters*, *Abandoned Engineering*, *Absolute Genius with Dick and Dom*, and a load more...

Even if you don't know the face, you'll have seen the work: she used to be the in-house prop maker for the Royal Institution, creating demonstration materials for (among others) the iconic Christmas Lectures. She's set off rockets with an aluminium finger, she's set fire to many, many things, and she's fond of making things explode. We caught up with her to find out how someone with a background in neuroscience ends up in engineering, what it's like to walk in Michael Faraday's footsteps, and why it's right to get things wrong.



Above
In Fran's hands, a big red button means that something is about to explode

HackSpace: Morning Fran! You have a background as a neuroscientist, but you've ended up exploding things on the TV. How did that happen?

Fran Scott: I think engineering was always in me: I just didn't know it. I went into neuroscience to get a Nobel Prize – you know, aim high! As part of my degree you could do a year in industry, and I was lucky enough to get a placement at the Australian Stem Cell Centre. Bear in mind, this was 17 years ago, so it was when we were just starting to tinker with them. And I was like – 'this is where the Nobel Prize is.'

But I was really frustrated. I grew up on a farm; I was quite practical. And then I went into a lab and it seemed so antiquated. I enjoyed the thinking; I enjoyed the problem-solving; I enjoyed the coming up with 'what ifs'. But I was frustrated by a lot of the processes. And so I decided that science wasn't really for me.

There was this thing that frustrated me a lot in science, which was that if you take something complex and explain it simply, then you obviously don't understand the complex parts of it. And this really perplexed me, because, to me, it's obvious that you need to understand something well in order to explain it. So anyway, a very, very astute careers advisor at my university asked me if I'd thought of science communication.

At this point there were two ways into science communication: you could go to Imperial College and do a master's in it. Or, you could get a job. I looked at the price of the master's degree and thought, oh well.

My friends took me to the Science Museum and I was like – whoa, what is this place? I'd visited museums as a kid, you know, my mum had dragged us to what was then the Museum of Film and Television in Bradford. So I knew these things existed, but not for science: science had always been put on this pedestal. Science was serious; science was not fun. And at the Science Museum, I was fascinated that you could combine these two things together.

I got a job there as an explainer, where you're in the interactive galleries, talking to the families and the schools and the parents as they come in, getting them to engage with the exhibits. And it's where I learned to communicate: a lot of people think that in science communications, the 'science' part is the hard bit, but it's really not. It's the 'communication' bit that's hard. Working at the Science Museum was a really great learning bed for that.

While I was at the Science Museum, I got back in touch with this love of making that I'd had as a kid on the farm where I would make things that we used.

And when I got promoted, and I was writing the shows and workshops, I'd go down to the workshop and try to make my own. I think I wanted this clock that, when you opened it up, springs all sprang into the air. And the guys in the workshop were great. I think I just needed that reassurance that this was OK.

> "My friends took me to the Science Museum and I was like – whoa, what is this place?"

HS: What do you mean by OK?

FS: Unfortunately, on my way to university, there's always been this sort of divide between design and technology on the one hand and the sciences on the other; if you could do the science bits, you didn't do design and technology, you had to just stick to the sciences to get the points to get to the university that you wanted to.

Now that I had my degree, I was still enjoying the design and technology. I started to build props [for the Science Museum] and got approached for lots of freelance work where people from other organisations were asking me to build things.

Back in 2020, I found out I was dyslexic. I had always worked very long hours as a student, without realising that other people didn't work the long hours; I'd be in the lectures, then I had to go back and sort of do the lecture again, so it made sense to me. I sort of assumed that everyone else had to do that – they don't.

And when I entered the world of work, and you started at 10am, and finished at 6pm, I was like: amazing! And so, I was filling the rest of my time with all of this other work, all this tinkering and just playing really. Whereas, with science, I felt that I never really fitted in, here I was finding this community, finding makerspaces. And it was OK to like these things. It was OK for things to not look perfect.

HS: That's ringing a bell. I've seen a talk of your video, on Ada Lovelace Day, where you say that one of your goals is for people to look at your work and say, 'That looks rubbish. I could have done that.' I love that. The Apple shiny, pure white cube is so alienating.

FS: This is why I ended up wanting to work for the Royal Institution. It's the home of science communication. It's where Michael Faraday was; it's the home of demos; this is the place to be. There's this corridor outside the theatre, but they call it the anteroom. And in there is a big oil painting of this eminent scientist doing a demonstration there. And the actual apparatus used in that demonstration is in a display case below it. And it's brilliant, because it's a Van de Graaff generator complete with bits of string that look like they're stuck on with chewing gum.

I remember seeing that and thinking, if something like that could end up in an oil painting at the Royal Institution, it →



Above ↑
Fran used to build props for the Royal Institution, such as this huge mobile representing the evolution of living creatures



Right
Engineering is messy – getting things wrong isn't just part of the process, it *is* the process

doesn't matter that my stuff is held together with sticky tape. And one thing that I learned, I suppose because I did a lot of work behind the scenes on TV shows, is that making stuff look pretty isn't the hard bit; the hard bit is making it work.

HS: What did making things for the Royal Institution look like? Because that looks like a dream job.

FS: It was. I was a northerner in London; I moved down with a suitcase, and I didn't know anybody. I remember walking into the Royal Institution, and, oh my gosh.... It's got this marble entranceway, with plaques everywhere and paintings. And I just remember thinking that they were going to throw me out for wearing trainers.

But they really welcomed me. There was a guy, Andy, who was in the demo team. Actually, at that point, he *was* the demo team. I became really good friends with Andy. I would be coming up with demonstrations, often involving explosions, in my flat and my small backyard. And you know, there's only so many times you can set off explosions in East London before you get a few calls.

I'd be like, 'Andy, I'm doing this demo; It might end up with some fire. Can I come to the RI to test it out, I'd like there to be another person there?' Some of my demos, I remember, I would call my mum on a video call so that she could watch it and call someone to come and get me in case it went wrong.

And so Andy would come up with his car, we'd pack in the equipment, he'd drive us to the Royal Institution on a Sunday, and we'd have a play – the deal was that if we got the demo working, he could then use the demo as well. This worked really well. It was nice to have someone to bounce ideas off. And there were other things, like I'd get invited on *Blue Peter* and poor Andy – he'd always be the workhorse at the RI – I'd be like, Andy, I could say that I need some help on *Blue Peter* if you fancy coming up for like the afternoon. It worked for both of us. But then Andy left.

At this point, I was running my own business where I was writing shows and taking shows internationally to coding and engineering shows, and his job became available.

I did not see that job becoming available in my lifetime, so I had to apply.

I'm so glad I did it. I don't work there anymore, but I'm so glad I did it, because I learned so much. And it gave me so much confidence. I'd always been quite hard on myself... when I was building things, no matter what schedule I set myself, I always seemed to be rushing up to the last minute and things wouldn't quite be working. I'd be like, Fran, why are you doing this to yourself?

> "I really wanted to get across that it may not work first time. That's not on you: that's part of the process. And it's a fun part of the process"

And then I realised that, even at the RI, that happens, and the RI has been around for 220 years, and if the RI does it, then maybe it's OK that I do it.

HS: I must admit, I only know the Royal Institution through the Christmas Lectures.

FS: The Christmas Lectures is the flagship that everyone thinks of when they think of the Royal Institution. And I realised that it's actually a very small part of what the RI does.

I suppose, just like with any other TV programme, it's underfunded, it's under-scheduled... there's a massive panic in making the demos. What you have got to build in the time allowed is unfathomable. And the team is incredible, how everyone works together. That was something that I had to learn as well: I'd been a freelancer, I had run my own business. I thought I knew teamwork, but I knew teamwork from a freelance perspective. And I have changed multifold since working at the RI in terms of how to work with people.

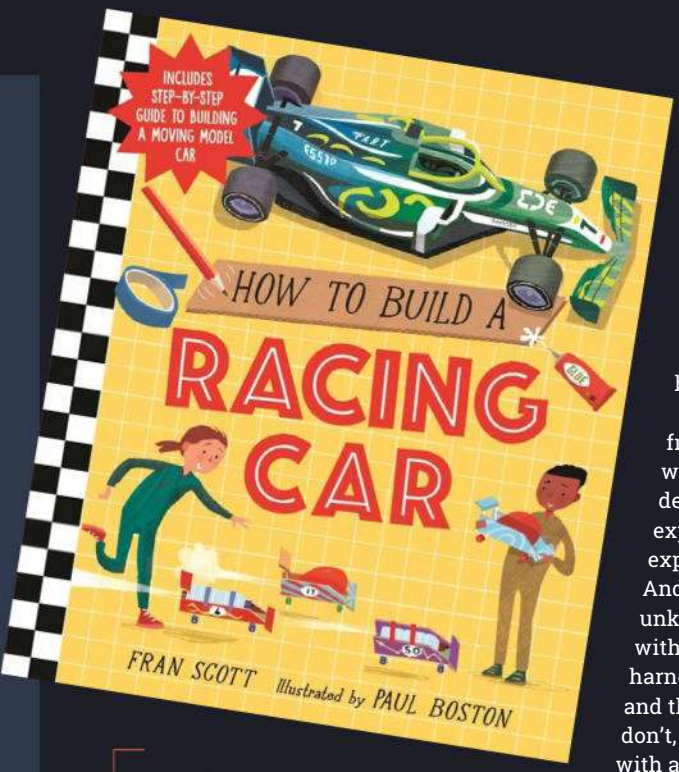
We worked so hard in those six weeks leading up to it. We wanted to make them the best we could be. And also for the lecturers, it's such a chance for them: it's decades of their work, and you want to help them explain it in the best way possible. But I also learned that having to do things last minute, that's not just me, that's just life.

HS: You've written a book: *How to Build a Racing Car* [due for release on 04 May]. What did you want to tell the world?

FS: What I really wanted to get across is that mistakes are all right. I had maker books as a kid, and it was actually recently I remember finding this book again full of things to do.

And it said on the front cover 'for boys' – I'd never seen that as a kid! I would try to make the things in the book, and my thing would not look like the thing in the book. I'd be like, 'Oh, I can't do this. I'm the failure.' And so, I really wanted to get across that it may not work first time. That's not on you: that's part of the process. And it's a fun part of the process. You can look at what didn't work and ask yourself, what if I do this differently? There's a testing and tweaking part of the book where it's like, how to make it faster, how to improve the traction.

And what I really wanted to try and get across was just fun. Not to 'make engineering fun', but just to show that it *is* fun. There's definitely some dubious humour in the book. There's not enough humour in science books. It just seems like science and engineering is put on this pedestal. What will you learn? Let's make this worthy. Or, you could just →



Above
***How to Build a Racing Car*: for engineers aged 7–11 (or older; we won't stop you reading it)**

have a play. We could laugh and make fart jokes.

There's scientific proof of the power of learning through play, and that doesn't have to stop when you're three or four years old. And so, I just really want to get across this essence of fun, of making mistakes, and the process, rather than, 'here's one step – here's another step, now do this, now here's your perfect thing – now why doesn't yours look like this?' It's more about the process than the product, I suppose. But the product itself is fun as well.

It will land differently with different people, you know – the brain is a wonderfully diverse thing. Some people do like strict rules. But this is why I sometimes believe that science is intimidating, and making is intimidating: because we're taught that there's a right and a wrong.

And we're told that if you can't get the right answer, then you must be stupid. If something doesn't work straight away, then it's on us. But that's so wrong. It's OK for something you're building to not work. That idea of playing, of making mistakes, of getting things wrong, isn't perpetuated enough. You know, don't cry when it doesn't work out. Just have a think and work it out.

And that's frustrating, but frustration is just an emotion where we don't know how to deal with what we're experiencing. And what we're experiencing is the unknown. And it's only by experiencing the unknown that we can come up with new things. So, we need to harness this feeling of frustration and the unknown. Because if we don't, we're never going to come up with anything new.

HS: That makes so much sense. Can you tell us about being a judge on *Lego Masters*? That looks like another dream job.

> I sometimes believe that science is intimidating, and making is intimidating: because we're taught that there's a right and a wrong

FS: *Lego Masters* was a lot of fun. I was brought in for *Lego Masters* for the second series; in the first series, they had the Lego judge Matthew, who was excellent. And then there was an expert judge, and that would change every episode in the first series. I was brought in to assess the builds from an engineering perspective. The whole reason I got into TV was because when I was growing up, I was always a bit annoyed that on factual programmes, the woman was always there to ask questions, while the man was always the knowledgeable one. And I'd be like, 'Oh, why does the woman not have to know anything? Why can't they be the expert?'

And so that's sort of why I made the beeline for being on TV. I'm an expert in science, I can talk; how hard could it be to get into TV?

What I found absolutely lovely was the people on the teams. It was a competition, but my God, it was *not* a competition. The contestants had to be held back from helping each other, because no one cared about winning: they were just so wonderfully happy to be in a room with that much Lego. And one thing I found lovely was that at the end of the first week of filming, we were like, 'Oh are you guys going for dinner?' And they were all going out to the Lego Store instead.

I really love people who are passionate about something. And if that something is engineering-based, even better. I really loved meeting the people and this was their passion, they loved it. What was even nicer was watching children meeting adults who still had that passion, and realising that it was OK to grow up loving Lego. They were getting this acceptance about who they were. They were only eight, you know, and saying, 'Oh, I can still like this when I'm an adult'.

There are interesting things with TV. I'm a dungarees girl, and I was having to dress up and wear dresses every week. And I remember I would get an ASOS delivery to work and there were some women there who knew about fashion. I'd get eight dresses delivered each week and get someone to tell me what to wear, and then seven of the dresses would go back. I know it's part of my job as a TV presenter to look presentable, but it's not something that comes naturally to me – I'd much rather be on the farm in my overalls.

HS: You've said that you aren't a science communicator, you're a science translator. Is that still the case?

FS: What I was trying to get across there – so that was probably about ten years ago, that I was saying, 'I'm not a science communicator, I'm a science translator'.

To me, science sometimes goes around and around in its bubble, and it talks in jargon. And everyone nods. And not everyone understands. But if you're nodding, then surely you're nodding because you understand. The thing about being a science translator was that I wanted to break that down and not use the jargon at all, but use words that everyone understands, and to actually explain concepts rather than just use the words.

But after that, I suppose I've re-evaluated my definition of communication. And it's because I used to think of communication as being broadcasting. Yeah, I mean, you communicate it out. And people will listen, maybe, maybe not. But now I define communication as something that only happens when the person who would want to receive information that you want them to receive actually received it. And if they haven't received it and understood it, then you haven't communicated; you've just broadcast.

HS: Oh, that's excellent. I always thought it was a weird thing that in order to get on at say, undergraduate level, you've got to learn the jargon. But then if you listen to the top, top experts in any field, they just don't use it all. They don't have to impress anyone anymore.

FS: It's like the classic thing that Einstein reputedly said: if you truly understand something, you can explain it to an eight-year-old. And working on children's TV, there have been times when I've been asked to talk about density, or magnetism, and thought yeah, I can explain this. And then I'm like, oooh, hang on. Give me a little bit of time here!

HS: Do you find there is a big difference between presenting for adults and presenting for children?

FS: No. The only difference is tone. And I suppose using jargon. What I would do differently when presenting to a child is I have a thing. I think it's getting better as I'm getting older, because I'm getting wrinkled and therefore I look wiser.

And so I have a thing of when I'm presenting, I have to prove my credibility. It's just a societal thing; people think that I don't look like I should know what I'm on about. So with that, when I'm presenting to adults, I will use jargon. And I will use jargon first, and then maybe explain it a little bit later. So you'd say, 'This thing is buoyant', and you would throw in the occasional jargonistic word just as this short cut to assumed knowledge.

And this is where you can use it in your favour. But I would never do that presenting to children. I sort of want children to think that they're smarter than me, because science is put on this pedestal for kids. So if they think the person presenting science is brilliant, that means they could be scientists.

Below
On stage, demonstrating a sensor that triggers an explosion via a Raspberry Pi when a glass of Baileys gets too full. As one does



FEATURE

IN THE WORKSHOP: Keeping an eye on nocturnal visitors

By **Andrew Gregory** @AndrewGregory83





I resolved to set up a wildlife camera

Above
If you feel the need to build a bird box, make sure you don't use pressure-treated timber – birds don't like it

Below
Woodwork is not only physically satisfying; it also smells good

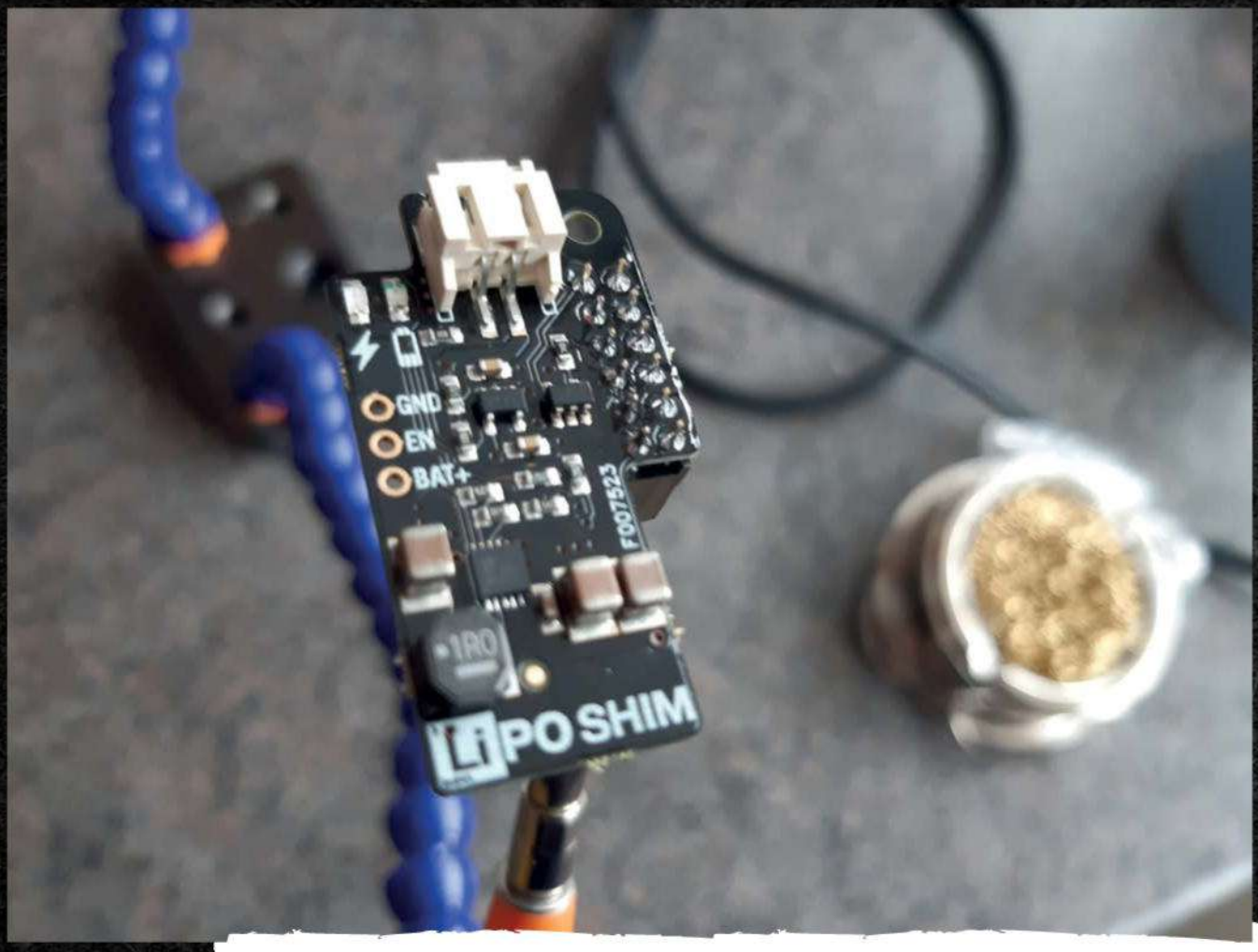


Something keeps waking my dog up in the middle of the night. That would be fine, except the dog keeps waking me up in the middle of the night. This is not fine. To find out what this is, I resolved to set up a wildlife camera to monitor the back garden.

There's a range of cameras on the market for Raspberry Pi: I thought about the Pi Camera Module 3, the Raspberry Pi Global Shutter Camera, the Raspberry Pi High Quality Camera, and various heat-sensitive modules. But as the visitor comes at night, the logical place to start was the night vision camera module available at Pimoroni et al.

I'm starting out with a Raspberry Pi 4, as that's what I have to hand; later on, when I've got the software working, I'll transfer what I've learned to a Raspberry Pi Zero W.

The camera module comes with two ribbon cables; one to fit a full-fat Raspberry Pi, the other to fit the slightly smaller camera connector of the Raspberry Pi Zero. There's a black plastic clip that pulls out, and pushes back in to tighten the connection. The blue side of the cable needs to face away from the pins on the connectors. In the terminal, I ran sudo apt update and sudo apt full upgrade, because it's nice to have the latest software versions, and even nicer to see loads of text fly past and feel like I'm some sort of hacker. Now, to test that you've got everything connected, run the following command: →



Above
The LiPo SHIM for Raspberry Pi Zero needs a bit of soldering, after which it Just Works

```
raspistill -v -o test.jpg
```

If all is well, you'll see an image flash up on your screen for a second or two. The colours will be off – that's because the night vision camera module uses infrared light (most camera modules filter this layer of the light spectrum out).

Now we know that the camera is connected, it's time to install the software. The package I went with is called Pi-Timolo (Raspberry Pi Time, Motion, and Low light), by Claude Pageau.

This is quite a hefty download, so run the following command:

```
curl -L https://raw.github.com/pageauc/pi-timolo/
master/source/pi-timolo-install.sh | bash
```

And put the kettle on while it downloads.

If you stuck around while Pi-Timolo was downloading, you'll have noticed the names of loads of packages installing. Fortunately for us, Claude has created a menu system to guide the user through configuring it all; we won't go into this here, other than to say that we keep to the defaults.

This object is going to live in the back garden, close enough to set up a video stream over the home Wi-Fi network, but far enough away that it'll need battery power. In hindsight, we should have gone for a USB battery pack – these come ready to plug into whatever device you're using, and have the benefit of some degree of protection. We opted for a 1200mAh 3.7V LiPo battery, which comes naked and fragile, and can't connect directly to the Raspberry Pi Zero W. We added a LiPo SHIM for Raspberry Pi Zero – this requires some soldering, but it comes with a header that you can solder the SHIM to, then slide it on and off the Raspberry Pi Zero W's headers so you can reuse the board in other projects.

That's the software and hardware just about sorted – now I just needed to build an enclosure. The classic for an outdoor project is a Tupperware box, with a hole cut out for the camera, but I want this to look nice, so I've found some scrap wood and I'm building a bird box-style enclosure. The benefit of this is that it uses a single plank of wood, cut into six pieces: a base, front, back, sides and a roof. The downside is

```
                pi-timolo Main Menu
Arrow/Enter Selects or Tab Key

a START    pi-timolo.py in background
b STOP     webserver.py - PID is 29772 http://192.168.1.172:8080
c SETTINGS Change Program Configuration Files
d PLUGINS  Edit Plugin Files
e CREATE   MP4 Timelapse Video from jpg Images
f CONVERT  Video from h264 to MP4 or Join multiple MP4 Videos
g RCLONE   Manage File Transfers to Remote Storage
h REMOTE   Manage pi-timolo using watch-app.sh
i UPGRADE  Program Files from GitHub.com
j STATUS   CPU temp=41.2'C   Select to Refresh
k HELP     View Readme.md
l ABOUT    menubox.sh
q QUIT     Exit This Menu Program

          <Select>              <Quit>
```

Left
Pi-Timolo's menu screen saves you from having to understand configuration files

Below
Raspberry Pi Zero W uses a different-sized power input to Raspberry Pi 4, and needs a different-sized HDMI cable if you're setting it up using a screen

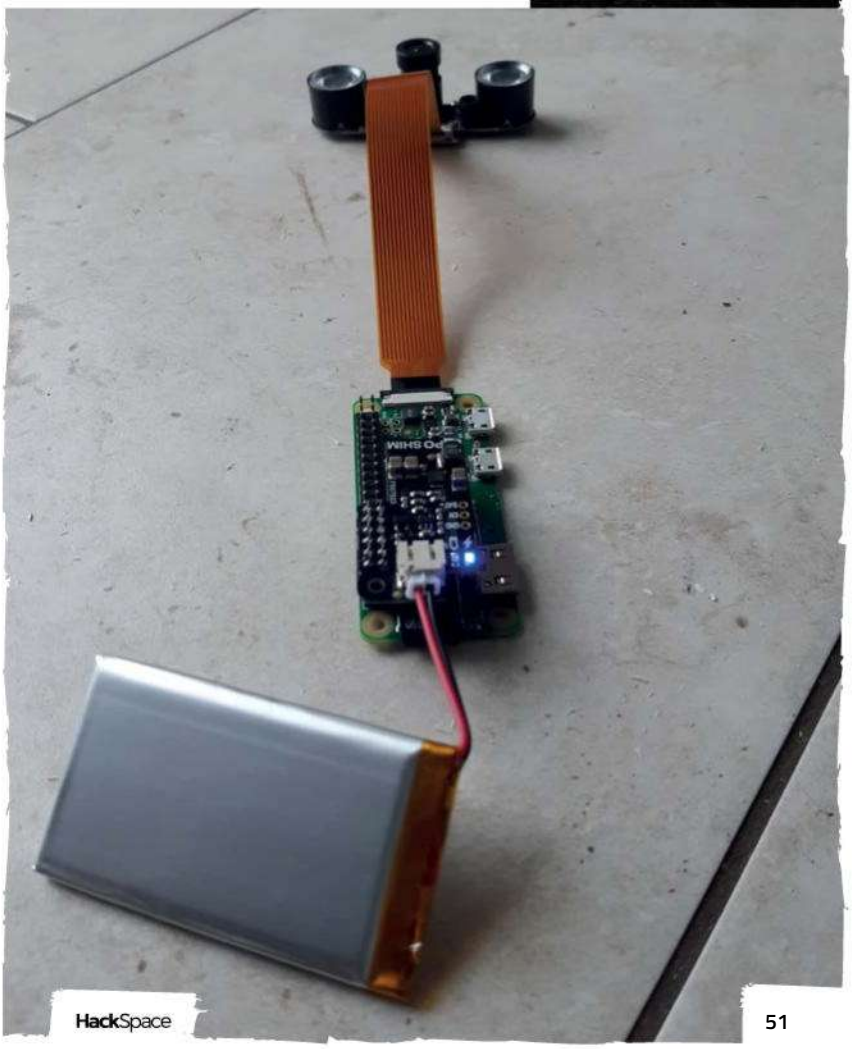
that if your wood has been sitting around warping for a couple of years, it might not fit together snugly.

For the hole that fits the camera module, we found a 20mm Forstner bit, which is the diameter that most closely matches the widest parts of the camera

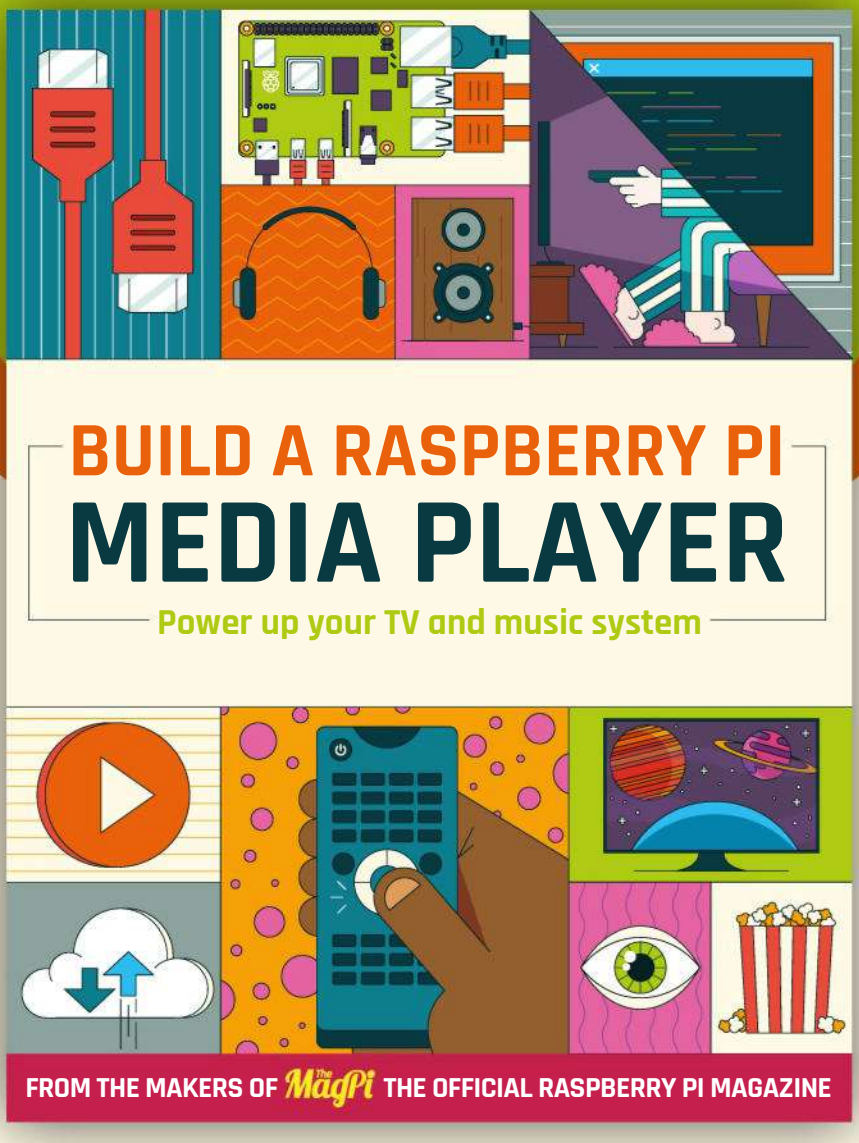
...and I'm building a bird box-style enclosure

module, bored out four holes in a line, then cleaned the hole up with a chisel.

What have we learned from this build? First, there's a reason that Tupperware containers are popular for outdoor projects. They're waterproof, naturally, but they're also thin enough that you can poke the camera lens through without interfering with either the light-dependent resistors or the infrared lights on the camera module. We've also learned that the 5MP sensor on the night vision camera feels tiny compared with the 12MP of the Raspberry Pi Camera Module 3, so we're strongly tempted to find some infrared LEDs and make our own camera module. ◘



Your FREE guide to making a smart TV



magpi.cc/mediaplayer

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects

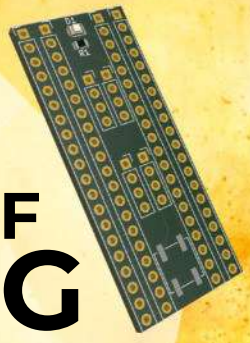
PG 60
PASTE SOLDERING
Let the oven do the work

PG 66

PAPER PLANES
Budget flying machines

PG 72

3D-PRINT NYLON
Engineering-grade prints

PG 54

SCHOOL OF MAKING
Start your journey to craftsmanship with these essential skills

54 KiCad

PG 78
AMSTRAD CPC
Strap a Pico to an 1980s classic

PG 80
SCOOTER
Electrify your wheels

Getting started with KiCad, schematics

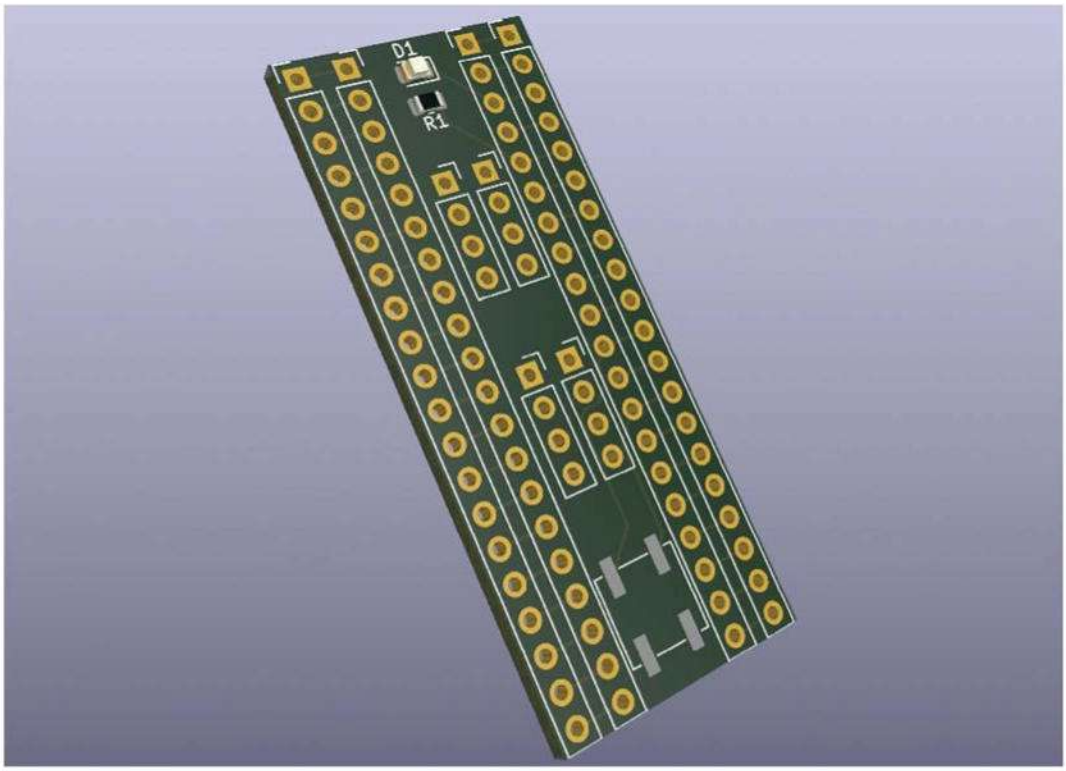
In this first of a series of articles around PCB creation with KiCad, let's dive into laying out a simple schematic



Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY space. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!



Above
The final simple add-on board for a Raspberry Pi Pico

KiCad is an amazing piece of free and open-source software that allows anyone, with some time and effort, to make high-quality PCB designs. Couple this amazing software with numerous PCB fabrication companies and even PCBA services – companies that will make and assemble your PCB designs – and there's never been a better time to get into this aspect of making.

PCB design can be a steep learning curve. With that in mind, we wanted to start this series of tutorials with a hack – this is HackSpace magazine after all! In the first two parts, we are going to create a PCB design to the point of getting it manufactured. However, to make this part accessible, we are going to cut some corners. Don't worry, we'll explain what the corner cuts are, and we'll do things more correctly in subsequent articles, once we have the basics down.

So, first thing, download and install KiCad – the current stable version is version 7, which was released in early February 2023. KiCad is available across a wide range of operating systems – Windows, macOS, and lots of Linux distributions. We're running it on Ubuntu, but it should be comparable across all the platforms.



Figure 1
The opening page of KiCad with the different applications that make up the KiCad suite listed

With it installed, click on the main blue 'Ki' KiCad icon to open the main application. You should see a screen that looks like **Figure 1**. You'll see that, really, KiCad isn't just an application, it's more a suite of applications that work together. Whilst you can jump into any application from here, the most common workflow for KiCad is to first work in the Schematic Editor and then, after creating a schematic, move to the PCB Editor to lay out the PCB physically.

Our first action should be to set up a new project. Click File > New to create a new project. It's worth putting projects into their own folder as each KiCad project generates around five project files and a folder in which it automatically generates some project backup files.

Once our new project is created, let's make a start. We're going to make an add-on board for the Raspberry Pi Pico. It's going to be a prototyping or 'kludge' board with not too many features on it for simplicity. It's going to have all the Pico pins broken out to through-hole pads, it will have a power-indicating LED, and it will have a reset button. After this, any spare area on the board will have simple through-hole pads on it to allow us to connect experiments to the board. As we said earlier, it's definitely hacked together, but will supply us with ample opportunity to learn some KiCad basics. To begin, open the Schematic Editor application by clicking the top icon.

> **It's definitely hacked together, but will supply us** with ample opportunity to learn some KiCad basics

You should now see a blank page ready for our schematic to be drawn (**Figure 2**). In the lower right-hand corner, you will see a small collection of text boxes which include various fields for the name of the sheet, the revision number of the schematic, and more. If you left-click somewhere on this section and then press the **E** key, you should launch a window called 'Page Settings'. You can edit this to add any text details or comments you want to add to this section, but you can also change the page dimensions – it defaults to A4 – and the orientation and more. Experiment inserting text into the Page Settings window to see where the text appears on the schematic.

With your page set up, we can now begin to add schematic symbols to the schematic. The most correct way to make a Pico add-on board would be to place a Pico in the schematic and connect everything to it, but we are going to use a workaround to do this in a much simpler way. The main idea of our target add-on board is for us to be able to solder in rows of header sockets so that we can connect it onto the pins of a Pico. In turn, we also still want to be able to solder to those pins, so we need each side row of pins to be broken out to another collection of pads. To do this we'll add two 20-pin connectors, one on each side of our schematic. To add the first one, we are going to click the 'Add a Symbol' icon – the third icon down on the right-hand column. The first time you click this icon, it may take a few seconds for the libraries of schematic symbols to load. Once loaded, you should see a window called 'Choose Symbol'. On the left-hand side of this window, you should see a list of items, each of which is a library of a group of schematic symbols. These are grouped in related items; so, for example, you might find,

QUICK TIP

If you hover over any tool icon in KiCad, you get a description of the tool. We'll use these tooltips to describe tool icons throughout these articles.

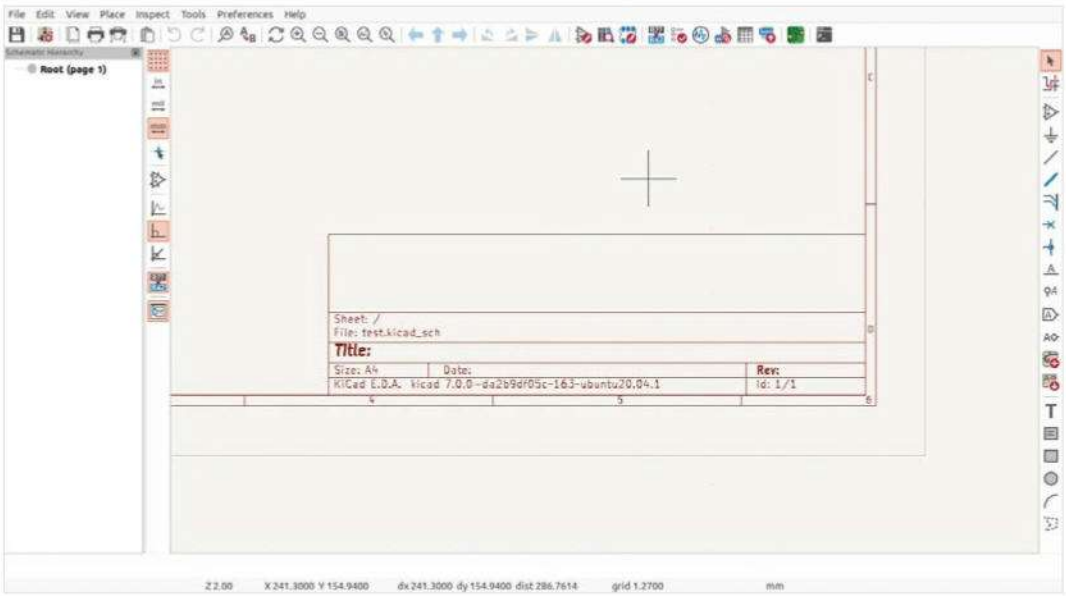


Figure 2
A blank Schematic sheet with text boxes for various schematic labelling and version numbering

KEYBOARD WARRIOR

Whilst you can do everything in KiCad with a mouse and pointer, it's really worth learning a few quick keyboard shortcuts that can help make life easier. Brilliantly, many keyboard keystrokes offer the same functions in both the Schematic Editor, which we are using in this tutorial, but also in the PCB Editor, where you will spend a lot of your time in the next part of the series. The first useful ones are the **F1** and **F2** keys for zooming. The **F1** key, when pressed, will zoom in, and **F2** will zoom out. Note that both of these zoom actions are centred around where your pointer is. With a little practice, it becomes second nature to move the pointer and zoom in and out to get the view you need. Next up are the **M** and **R** keys. If you select an object in the Schematic Editor or the PCB Editor, you can press **M** and then that object will move until you left-click to place the object again. Note that in the PCB Editor, you can select smaller parts than the complete footprint of a component. You can therefore select and move silkscreen labels and more. If you want to select the whole component, selecting a component pad will usually then select the entire footprint. With the **R** key, you can rotate a selected item in either the Schematic Editor or the PCB Editor, again single left-clicking to place the part when rotated correctly. Note that you can rotate the item repeatedly to get to the orientation you require. Finally, another useful shortcut is the **E** key. With an item selected, pressing the **E** key allows you to edit that item's properties. This can be any number of editable parameters, from labels to pad sizes to hole dimensions and more.

QUICK TIP

Whenever a tooltip has a letter in brackets, that letter is a hotkey to switch to that tool.

at the top of the list, '4xxx' which, if you click on the drop-down menu button next to the name will reveal a library full of the CMOS 4000 series of logic chips. You can of course manually scroll through the libraries and look through the items they contain, but you can also search the libraries to find what you need. We are going to add a connector symbol that represents the 20-pin header we eventually want to be able to fit. To do this, type 'conn' into the search bar and, as you type, you should notice the list items are now all the items that start 'conn'. Scroll down the list and select the item that reads 'Conn_01x20_Socket', and then click the OK button in the lower right-hand corner of the window (**Figure 3**). The Choose Symbol window will close and you will be back in the Schematic Editor, and you should have a symbol for our 20-pin socket attached to your pointer. Move the pointer to where you want to place the connector and left-click to place it.

A SACKLOAD OF SOCKETS

We are going to place three more of these into the schematic. We can again click the Add a Symbol tool icon but notice that, as the library list area populates, we now have a 'Recently Used' area at the top with our 20-pin connector listed underneath. Click this listing and then OK, and we can place the next connector in the schematic. Repeat this until you have four of the connectors placed. Next, use the **M** and **R** hotkeys to move and rotate the connectors so that we have two pairs of connectors in line with each other, with the small circle ends facing inwards

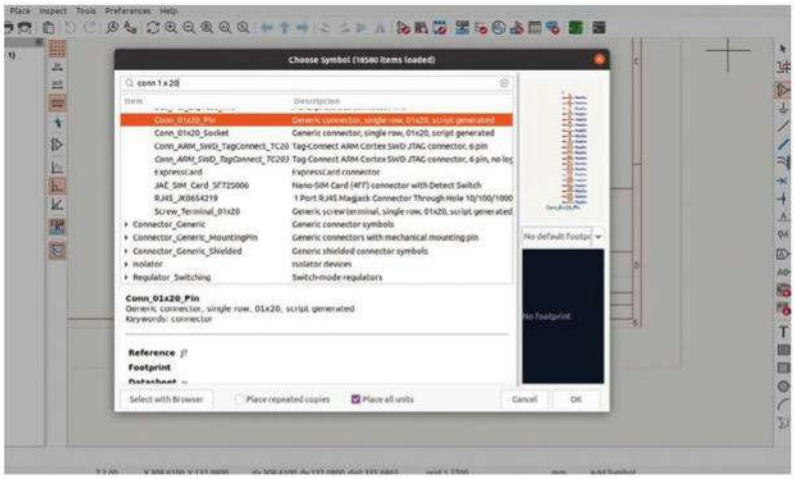


Figure 3 The Choose Symbol dialog that allows you to search for schematic symbols

towards each other. You can also right-click on a selected symbol and use the Mirror Horizontally tool. The small circles are the part of the symbol to which we will create connections.

Using the Add a Wire (W) tool, you can left-click on one of the connector pin circles and then draw a connector wire line across to the opposite connector pin (**Figure 4**). If you make a mistake, you can use **CTRL+Z** to undo the last action – or, to cancel the wire mid-draw, you can right-click and select Cancel from the drop-down menu.

Continue to wire between each of the opposite pins on each pair of our connectors until they are all connected together. Notice that each of the connector symbols have a couple of text references connected to them. It can be good practice to edit these so that they are useful and help you keep track of what things are. Select the whole of one of our connector symbols (use the selection tool to draw a box over the entire symbol) and then press **E**. You should now see a Symbol Properties dialog box (**Figure 5**). You can now edit the 'reference' (if required) and the 'value' text entries. We edited ours and gave each connector a descriptive 'value' such as 'Pico_Pins_1-20_Left' etc. This is useful when we lay out our PCB in the PCB Editor, as it ensures we can identify the multiple connectors correctly.

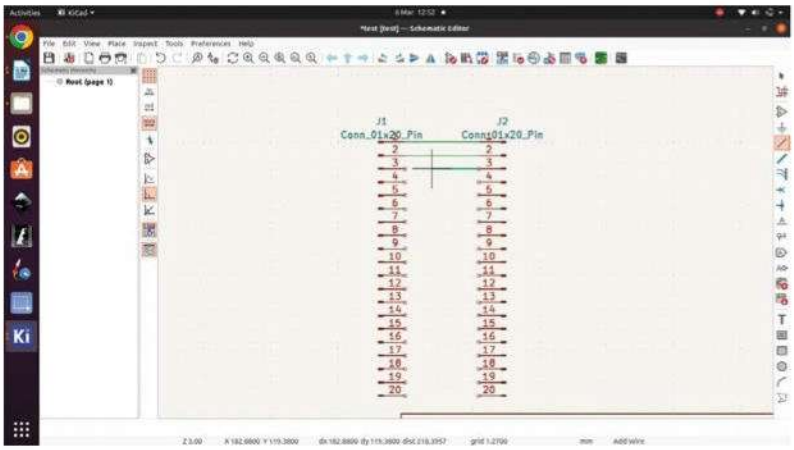


Figure 4 Adding our wire connections between the pins of the schematic symbols

> To finish our schematic, we are going to add more **components and make more connections**

To finish our schematic, we are going to add more components and make more connections. To begin, let's add a resistor and an LED and connect them to pins 38 (GND) and 36 (3V3 (OUT)) on the Pico. These pins are the third pin and fifth pin down on our right-hand connector. Use the same techniques we used earlier to choose a symbol, searching 'R' for a resistor and 'LED' for an LED symbol. Add wires to connect the circuit together, as shown in **Figure 6**. Next, let's add a switch symbol which we will wire in as a reset button for the Pico.

Adding a reset button is a good example of a way that KiCad works that some other electronic design automations (EDA) don't. In some EDAs, the symbol you choose at the schematic level defines exactly the hardware package of the electronic component that will be on the PCB. For example, in other packages you wouldn't place a generic resistor symbol, you would place a resistor symbol linked to a specific resistor, say a 6 mm long 1/4 watt carbon resistor placed horizontally. This would mean that, if we wanted to change the component on the PCB, we have to change the schematic. In KiCad, the schematic symbol has to be assigned a footprint – this is why we are going to place a symbol for a single-pole single-throw switch (SPST), but the actual component can be any SPST switch or any →

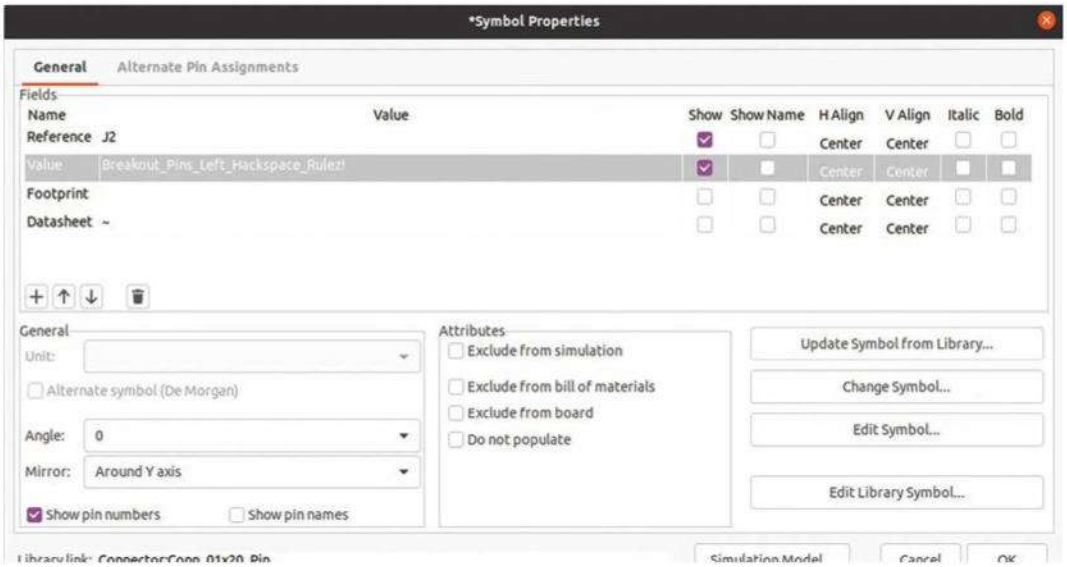


Figure 5
Editing a symbol's properties allows us to give symbols more meaningful names

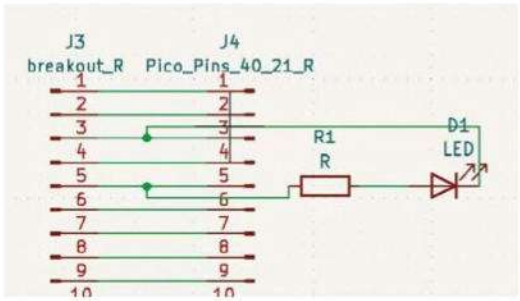


Figure 6
Adding a resistor and LED to our schematic

button package. In our case, we'll choose some type of momentary push-button for the reset. This way of working means that if you find you need to replace a component with a different type (commonly as you can't find a component in stock), you simply change the footprint associated with the symbol and don't have to edit your correct schematic. It also makes the schematics concise and readable, which is important if you want to share your design.

SWITCHED ON

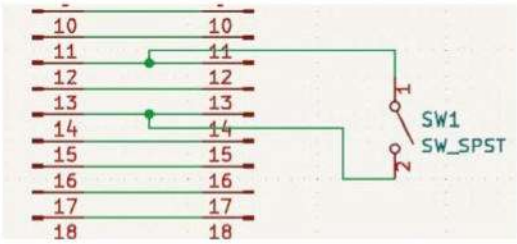
To add the switch, search the Choose a Symbol dialog with 'sw_spst' to take you directly to the single-pole single-throw switch and again connect wires, as shown in **Figure 7**. Finally, add four connectors to act as our prototyping areas on the board; search for 'conn_01x04' to take you directly to this symbol.

Let's now set up the associated footprints for each schematic symbol. Click the Run Footprint Assignment Tool icon. Similar to the Choose Symbol dialog, a window will appear with a list of footprint libraries down the left-hand side, a centre section with the schematic symbols of the project listed, and a right-hand column of filtered footprint results. You can see a collection of three icons at the top of the window called 'Footprint Filters' and, in the first instance, make sure all these are selected. As you get used to component filtering, you might find you prefer to use different combinations of these filtering tools (**Figure 8**). Highlight in the centre section one of our 20-pin connector components. We need to find a footprint that has the same number of pins and has a footprint of a through-hole pad for each pin. As the Raspberry Pi Pico pins are spaced at the common 2.54 mm pitch between each pin, we also need to make sure our footprint is spaced similarly. You should

ADDING PARTS

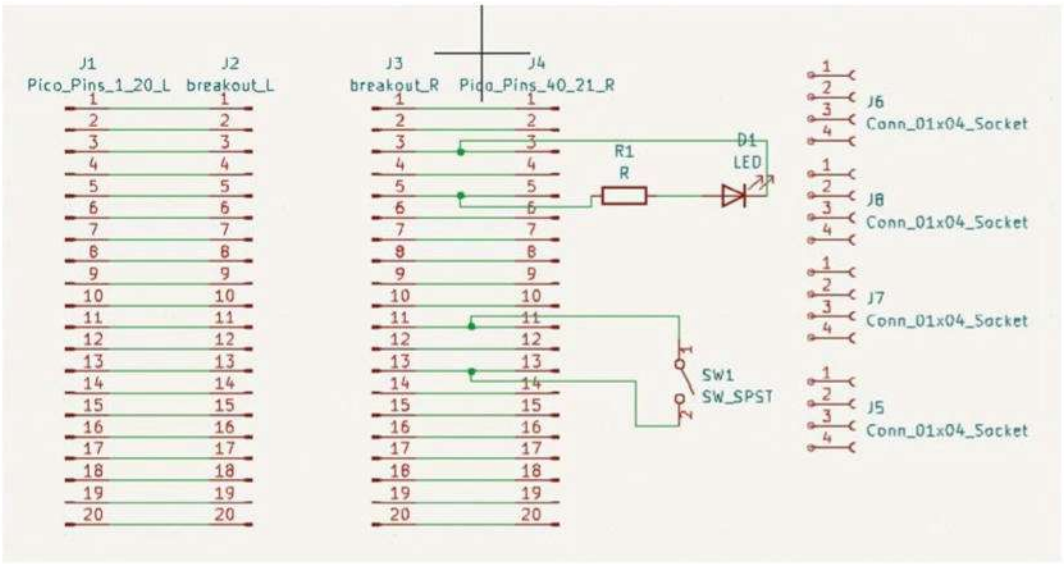
As you can tell from this tutorial, KiCad uses Schematic Symbols and Component Footprints to create schematics and PCB designs. You should have a lot of standard libraries built in, and everything in this project is just using these libraries. Of course though, you can create your own schematic symbols, component footprints, and even 3D models of components and incorporate these into your own designs creating custom libraries. We'll cover this, and lots more in future sections.

Figure 7
Adding a single-pole single-throw switch to the schematic to work as a reset button for the Pico



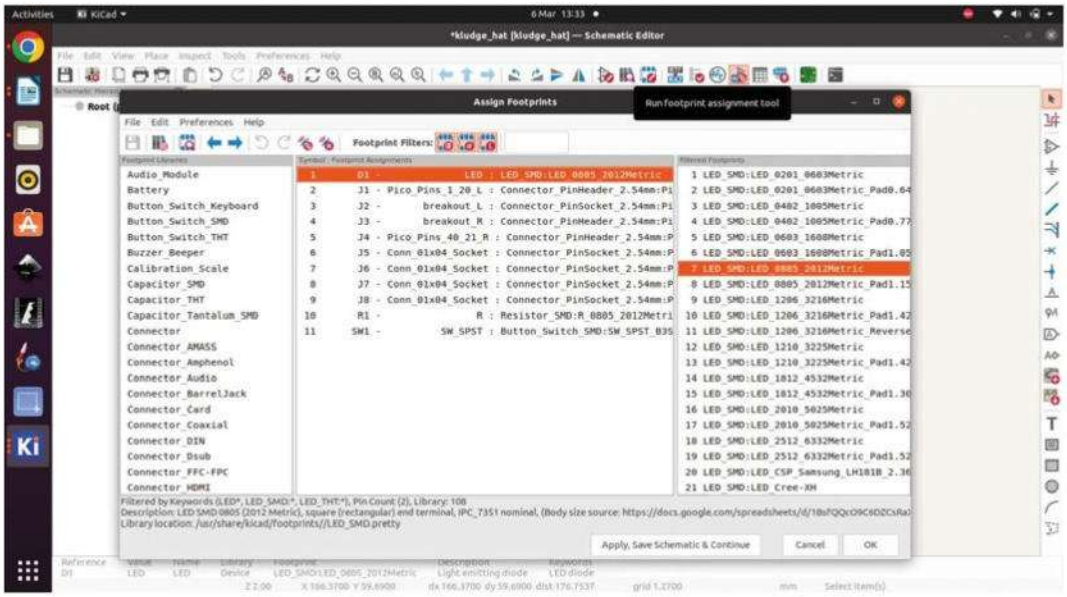
QUICK TIP

Obvious, but remember to press the Save button now and again to save your work!



have a filtered list on the right-hand side – it should be filtered to only contain items that would attach to the 20-pin connector symbol. We chose item number 33 from the connector footprint library, which reads 'Connector_PinSocket_2.54mm:Pinsocket_1x20_P2.54mm_vertical'. A single left-click on the item highlights it in the list, and you can use the 'View the selected footprint in the footprint viewer' tool icon to open a window showing the PCB footprint layout design to check if it looks correct. Double-clicking the highlighted footprint should then add that footprint name opposite the schematic symbol in the central list. Continue and add the same footprint to all four connector symbol listings, although we'll discuss later why that isn't totally correct, and then click the 'Apply, save schematic and continue' button. We've decided to add surface-mount components for the LED, resistor, and the button, although we have gone for SMD package sizes and footprints that would enable us to hand-solder these. For the LED, we chose the LED_SMD:LED_0805_2012Metric footprint; for the resistor, the 'Resistor_SMD:R_0805_2012Metric'; and for the switch/button, a 'Button_Switch_SMD:SW_SPST_B3SL-1002p' component. For the detached 4-pin connectors, we simply selected 'Connector_PinSocket_2.54mm:PinSocket_1x04_P2.54_Vertical' for each connector. With all those footprints assigned, click 'Apply, save schematic and continue'.

In the next instalment, we will import the list of component footprints into the PCB Editor and physically lay out our board design ready for fabrication!



Above
Our complete circuit schematic

Figure 8
The Assign Footprints dialog

TUTORIAL

Surface-mount soldering with paste

Ditch the soldering iron; melt the metal with an oven



Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Last month we looked at surface-mount soldering. We used a soldering iron to attach small components to the top of a PCB. However, this isn't the only way to attach surface-mount components. While soldering irons can be great for small jobs or one-offs, they're a bit slow if you need to attach a lot of components, especially if they're tiny surface-mount components. It can be much faster to use solder paste – this is a gel made up of small balls of solder suspended in a flux. You put dollops of the paste on the pad, pop the components on top, and then heat the lot up so that the flux burns off – the solder melts everything and joins together. Let's take a look at how to do this.

There are a lot of options here. There are different ways of applying the paste, and different ways of heating the board. This issue, we're going to look at a manual method of applying the paste – we'll look at using stencils next issue.



Right
Our 18 gauge needle was a little too big for the 0402 components, but they did still all solder correctly

You can apply paste in a couple of ways. The easiest option is to get a dispensing tool. These work in a similar way to the tools for squirting silicone sealant or decorators' caulk, just on a smaller scale. Dispensing tools only work with specific sizes of solder syringes, so make sure you get the right option.

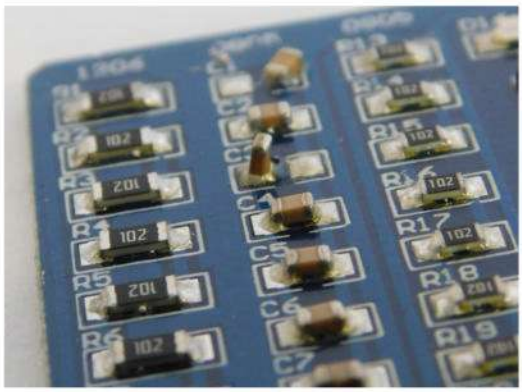
You can also dispense from a syringe by hand, or you can apply a dollop of solder paste using a toothpick or similar small pokey object. However, we'd strongly recommend getting a dispensing tool – they're fairly cheap and much easier than trying to do it by hand. If you do choose to go down the manual route, the same basic steps apply.

If you're dolloping it on using a toothpick, we'd recommend mixing in a little extra flux (just about 20% or so) to the solder paste. This makes it easier to work with, and it sticks to the pads better. Don't add too much, though, or the solder runs all over the board when it heats up.

There are quite a few different formulations of solder paste. We would strongly recommend using lead-free solder paste. The stuff can get on everything, so it's better if it's not a neurotoxin. Lead-free doesn't mean non-toxic, though, so still wash your hands thoroughly after using it, and take the usual precautions when handling it.

PADDING OUT

The first step is to get paste onto the pads. The paste isn't always very good at sticking to the pads, so you might have to give it a bit of a push to get it on. You don't need much solder paste. With surface-mount soldering, you're not trying to fill a gap in the same way you are with through-hole, so you



Above
Tombstoning is where components rise up with surface tension – you can fix it with hot air or a soldering iron

really just need a small drop to make the connection between the component and the pad.

If you're dispensing the solder from a syringe, the nozzle will limit the size of the drop you can give. Large nozzles make large dollops, and small nozzles make small dollops (OK, small nozzles can do large dollops, but it's slow and requires you to squeeze a lot). We'd recommend gauge 18 or 20 nozzles for work around 0805 or 0603.

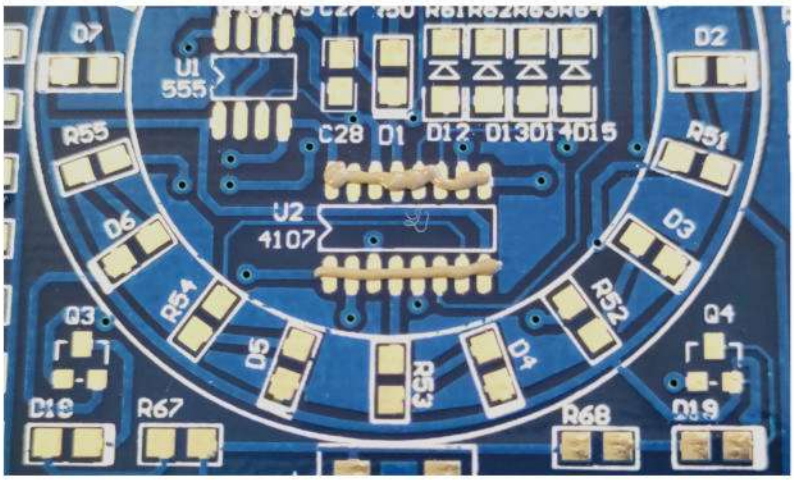
You don't need to be perfect with this. Solder paste has a lot of flux in it, so this will help the solder find only the pads. For finer-pitch components, such as chips, you can just run a bead of solder across all the pins in one go, and it'll find its way to the pads once it's heated up.

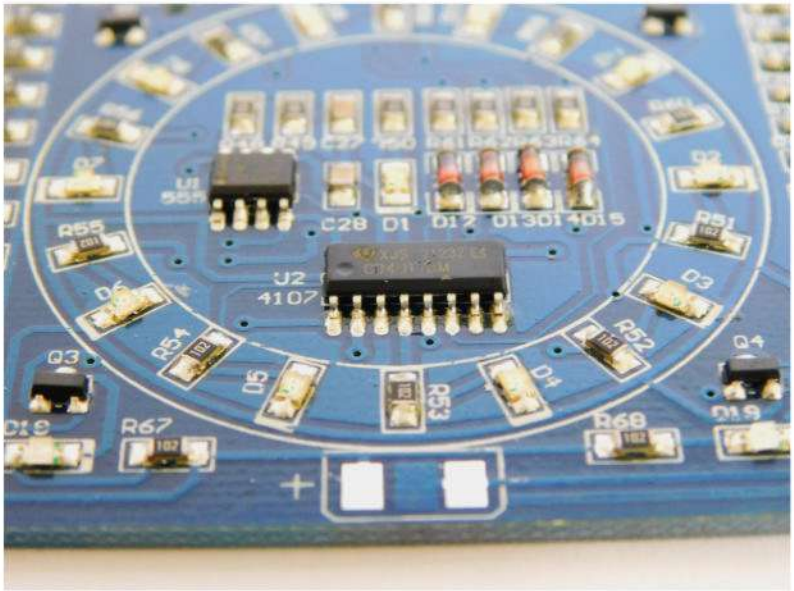
> Lead-free doesn't mean non-toxic, though, **so still wash your hands thoroughly after using it**

If you make a mistake, you can wipe it off with a tissue, so just have a go and wipe off if you need to.

Once you've got paste on some (or all) of the pads, it's time to populate them. This is just a case of grabbing your tweezers and placing the components on their pads. The solder paste should hold them in place, but if it doesn't, you might have to add a bit more.

At this point, you might be wondering what you do if there are components on both sides of the board. The answer is … it's tricky. There are a few →



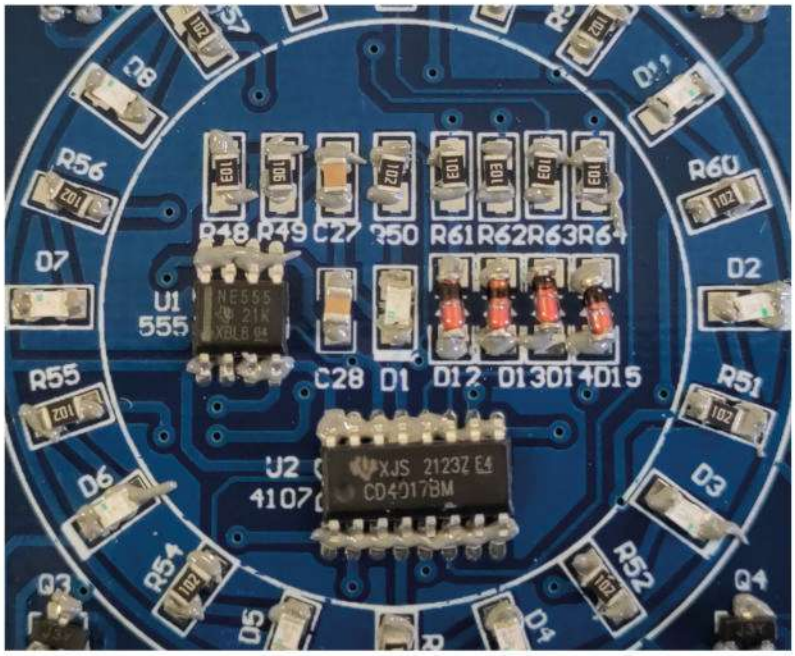


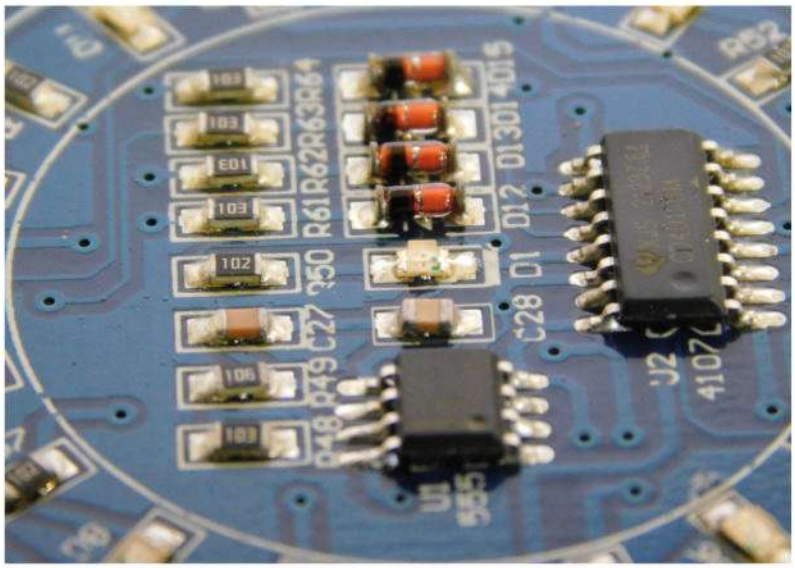
NEEDLES

There's a wide range of needles available for soldering with. We like to use around 18 to 20 gauge for most work. As well as the familiar steel needles, there's also tapered needles and Teflon-lined options.

There are also many types of solder paste and applicators – different people prefer different options. It might be worth getting a selection pack to let you try out some of the options before committing to a large pack. That said, if you're only going for one job, we find 20 gauge needles a good starting point. Either metal or tapered should work. Needles and syringes typically use Luer locks to attach, and this is fairly standard between manufacturers.

Above
For fine-pitch connectors, you don't have to get the paste exact, you can just run it across a row of pads and the solder mask will ensure it ends up in the right place





Above
This might look a little messy – and it is a bit – but all the components soldered correctly



Above
A dispensing tool makes applying paste much easier

solutions, but none of them are great. You can use different temperature solder pastes: so you first do one side of the board with a high-temperature solder paste, then heat that up and solder it, then do the other side with a low-temperature solder paste. To be honest, we almost always do one side with paste and the other side with a soldering iron, and try to minimise the number of components we have to solder with the iron.

> "The price of purpose-designed tools has **dropped significantly in recent years**"

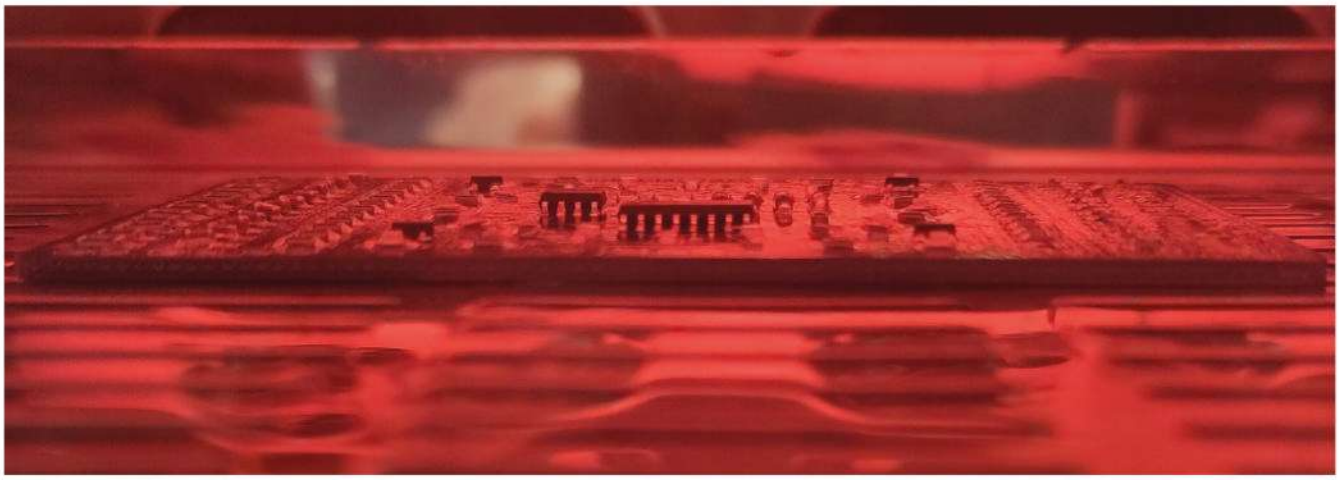
Once you've got the parts added, it's time to heat the board. The best thing to use is a reflow oven, which will heat the board evenly. However, this is a bit of an investment if you're only planning on doing it occasionally. The second best option is a hotplate. This is exactly as it sounds – a plate that heats up to a specified temperature to melt the solder. The final option is a hot air gun. If you're using a hot air gun you must be very careful with your air speed, otherwise your carefully placed components will be distributed across the room.

It used to be common to hack together a PCB heater. People used skillets, upturned irons, and all sorts of other bits and bobs to heat up the solder. The price of purpose-designed tools has dropped significantly in recent years, so hacking together your own tools isn't quite as attractive as it once

SOLDER TYPE

Different brands of solder paste differ from others in a multitude of ways. Some stick to the pads better, some are runnier, and some are thicker. Unfortunately, there's no easy way of telling the difference without trying it out.

There is, however, one parameter that most solder pastes give – the 'type'. This corresponds to the size of the solder particles in the paste. You'll probably come across type three and four pastes, with higher numbers being smaller particles. Type four should work with most needle gauges larger than 18, so is a good bet.



was. That said, you can still save money by rolling your own – especially if you can scavenge a heater from somewhere.

HOW HOT?

Solder paste has specific heat profiles that are designed to get the best out of them. This usually involves heating the board up to around 150 °C for a few minutes to ensure everything is up to temperature (this is known as a 'soak'), then ramping the temperature up to the full soldering temperature for a minute or so to 'reflow'. This profile allows the solder to melt without putting too much thermal strain on the components. You generally want to minimise the amount of time the components are at the full temperature, and by preheating everything, you only have to have a little blip at full temperature. Your solder should have a technical data sheet containing details of what this profile should look like.

This heat profile is the ideal. How close you can get to this depends on the equipment you have. Most of the time, you'll get away with ramping the temperature up to the full soldering temperature and waiting for the solder to melt.

When the solder melts, it will adhere to the pad and the component. Surface tension in the liquid metal will pull the component into position. Most of the time, this will be the correct position, but occasionally it will go wrong. This can mean components rotated 90 degrees sideways, or even 'tombstoning' by pointing upwards.

The big difference between using a reflow oven and the other options is that a reflow oven is completely enclosed. This means that it can heat the board evenly, and you have a good deal of control over the temperature. However, it also means you can't poke the board if things aren't going as planned.

If you're using a hotplate, a little poke with the tweezers can sort out any problems. If you're using a reflow oven, you'll have to wait until the board has cooled, and then attack it with either hot air or a soldering iron.

All that's left is to test your circuit.

Soldering with paste can be much quicker than using an iron, especially if the parts are small. It can be useful if you're building more complex boards or doing short assembly runs. It's not particularly difficult, but does require a bit of practice, and needs a little specialist equipment. It's a great skill to have in your maker armoury.

Above
The PCB baking in the oven

Below
An enclosed heater gives the most control over the heat, but soldering hot plates and hot air also work



the

COMPUTERS THAT MADE BRITAIN

*"**The Computers That Made Britain** is one of the best things I've read this year. It's an incredible story of eccentrics and oddballs, geniuses and madmen, and one that will have you pining for a future that could have been. It's utterly astonishing!"*

- **Stuart Turton**, bestselling author and journalist

Buy online: **wfmag.cc/ctmb**

Available on amazon

OUT NOW

the COMPUTERS THAT MADE BRITAIN

THE HOME COMPUTER REVOLUTION OF THE 1980s

TIM DANTON



TUTORIAL



Advanced paper aeroplanes

Whilst the art of simple paper aeroplanes is commonly known, let's look at getting a little advanced with our designs



Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY space. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

Above A collection of paper aeroplanes

Paper aeroplanes have always remained popular – pretty much everyone has had a go at making one once. They are the simplest and the cheapest way to experience making something that flies, and that cheapness, a cost of just a single piece of paper, often leads to people being creative and experimental. They also are really quite helpful in terms of learning the art of 'trimming', which is applying the little twists and tucks to any model aeroplane to get it to fly in the manner you want it to – a great skill to practice if you want to move on to building more complex flying crafts. We'll start with the basic dart-style paper aeroplane; you've likely made one of these in your life, but let's walk through the process just in case.

Begin with a sheet of A4 paper. In portrait mode, fold it in half along the longest axis, flatten the fold, and then score the fold by running a fingernail along it. Next, unfold the fold you just made.

Next, take the top corner of one side of the page and fold that corner over until it touches the centreline fold you just made. Again, flatten this fold and crease/score the fold with a fingernail. Repeat this fold on the opposite side (**Figure 1**). Next, we are going to fold each of the side triangles we formed in the last step in half again. Take the angled edge of a side and fold it over so that it aligns with the vertical centre fold, then crease/score it flat (**Figure 2**, overleaf). Repeat this for the other side.

Next, fold one half of the sheet over onto the other half using the first centre fold line you initially created. The edges of the angled sections should meet evenly. The final step is to fold down the wing; take one side and fold the wing outwards and over to meet the bottom edge, which is the centreline fold (**Figure 3**, overleaf). Crease the wing fold so that it remains folded, turn the aeroplane around, and repeat to fold down the other wing. Keeping hold of what is now the fuselage (the area that isn't the folded down wings), you can bend up each wing until, to begin with, it is at 90 degrees to the centre fuselage section. Repeat on both sides.

You have now created a dart-style aeroplane! Most darts need a reasonably good throw to get them to fly, and they tend to soar quite quickly. You can experiment with a few different trimming

techniques to change the flight characteristics of the dart. Often a dart will roll as it flies, this is where the wings spin, through the aeroplane being upside down and the right way up as it travels forward. A way to reduce this is to increase the dihedral angle of the wings. Adding dihedral means that you increase the angle of the wing from the fuselage so that, from the rear, the aeroplane appears to be more of a Y shape than a T shape. Experiment with this wing position and see what results you get!

As a child, I had a copy of the excellent *Advanced Paper Aircraft Construction Mk II* book. It contained all kinds of tantalising and technical paper aeroplane designs – they often had challenging origami-style folds that, whilst excellently drawn, were sometimes difficult to comprehend from a diagram. One design that uses some similar complex folds is a glider with landing gear or legs (**Figure 4**, overleaf). As it's a little tricky to describe, I've uploaded a quick video tutorial showing how to create this aeroplane, fold by fold, in case you get stuck in the next section!

> "You can experiment with a few different trimming techniques **to change the flight characteristics of the dart**"

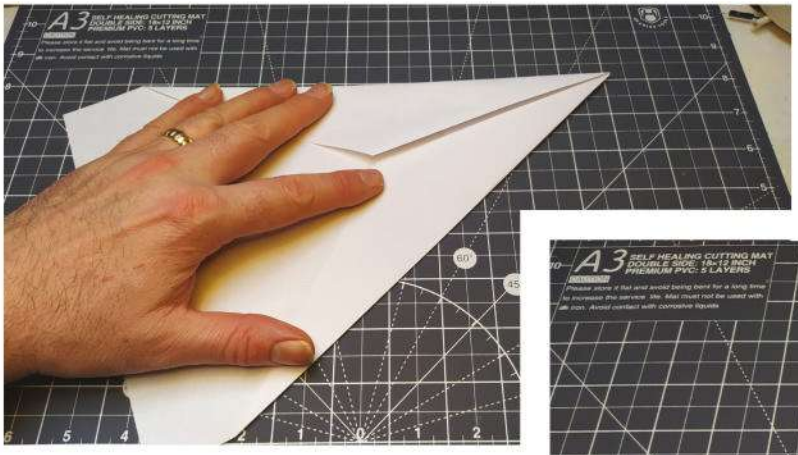
Fold a sheet of A4 paper in half, in portrait orientation, and then unfold it. Next, take the top right corner of the page and fold it over all the way to the left-hand side, so that you create a triangular shape coming to a point at the top left corner. Then, unfold that triangle and repeat the fold on the opposite side and unfold. Next, place your forefingers under the page, either side, halfway down the edges between the top corners and the end of the triangular folds you just created. You're going to pull the paper towards you and inwards – you should see a triangle shape begin to appear above your fingers. Fold and score the upper triangle flat to the page (**Figure 5**, overleaf). Again, if that last section lost you, refer to the tutorial video to see the fold created.

Along the bottom edge of the newly formed triangle, you should now have a stack of three layers on top of the original sheet. Insert your thumbs in the middle of the layers so that you have two layers of paper above each thumb. You're going to fold over a section by rolling your thumbs over towards the pointy end of the construction, folding over around 15–20 mm. As you roll this over, you will have some curved sections that rise up in the middle of the design. As you score the bottom edge of the fold, push these sections flat. If all has gone well, you should now have something that looks like **Figure 6**.



Figure 1
Step 1. Hide at the back of class, and make sure teacher can't see you

TUTORIAL





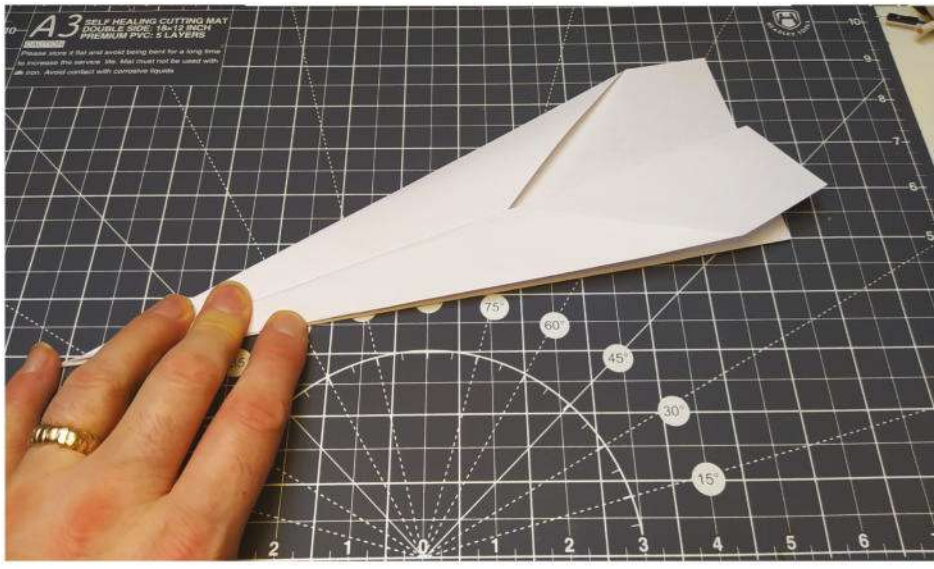


Figure 2
Folding the sides over to make the pointed, classic dart shape

Figure 3
Folding the wings down to complete the simple 'dart' paper aeroplane

The complex part of this construction is over. To finish the aeroplane, fold the nose down to meet the centre of the complicated fold you just made, then fold the entire aeroplane in half along the centre fold you made initially, keeping the complex area that will fold into the landing gear on the outside. Fold down the wings to 90 degrees, using the edge of the small triangles in the centre as a guide. You can then put two 90-degree folds into the landing gear to put it into place. We've found that adding some vertical stabiliser sections, by folding up 10 mm or so of the end of each wing, helps stabilise this aeroplane in flight, and if you create some small upward creases evenly in the back of both wings as elevators, you can get it to perform a decent loop!

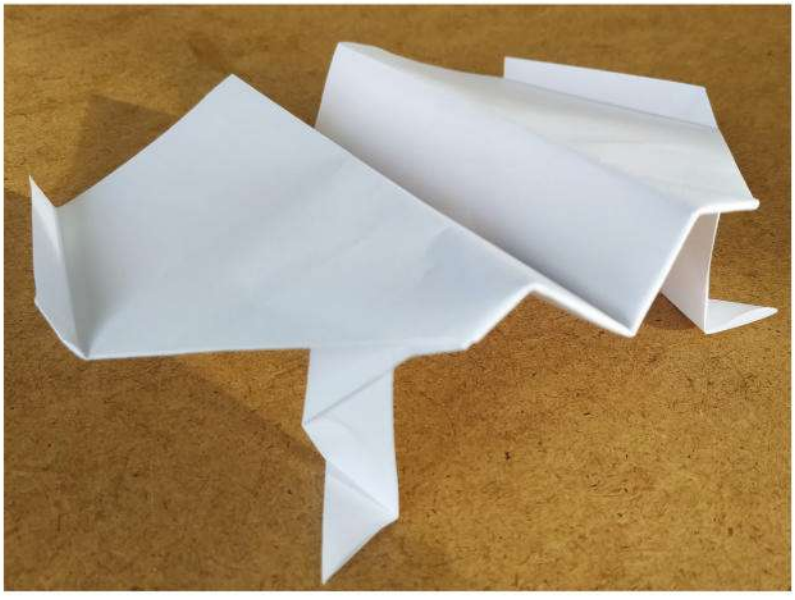


Figure 4
The 'landing gear' aeroplane is a great design that flies and lands well

Further research into advanced paper aeroplane construction led to the discovery of two interesting older books (**Figure 7**, overleaf): *The Great International Paper Airplane Book* and *THE Paper Airplane Book: The Official Book of the Second Great International Paper Airplane Contest*. Both books were published following international paper aeroplane competitions. These titles can be found second hand, but they are also available to loan digitally via **archive.org**.

Both books feature many fascinating designs pulled from the winners of the different classes of paper aeroplanes flown at these competitions. As you will see, one of the books is from a competition in 1967, while the other is from 1985. Both competitions saw people from all over the world mail their designs over to compete in various classes, such as distance, flight duration, acrobatics, and more. One interesting difference between the books and the competitions is that, for the later competition, the rules were changed to allow for laminated paper aeroplanes. These were a new type of paper aeroplane to me – I was interested to see how well they flew. The premise of the laminated aeroplanes is simple: you cut out various flat panels from thick paper or thin cardstock and then laminate them together with glue. In fact, the later book has a

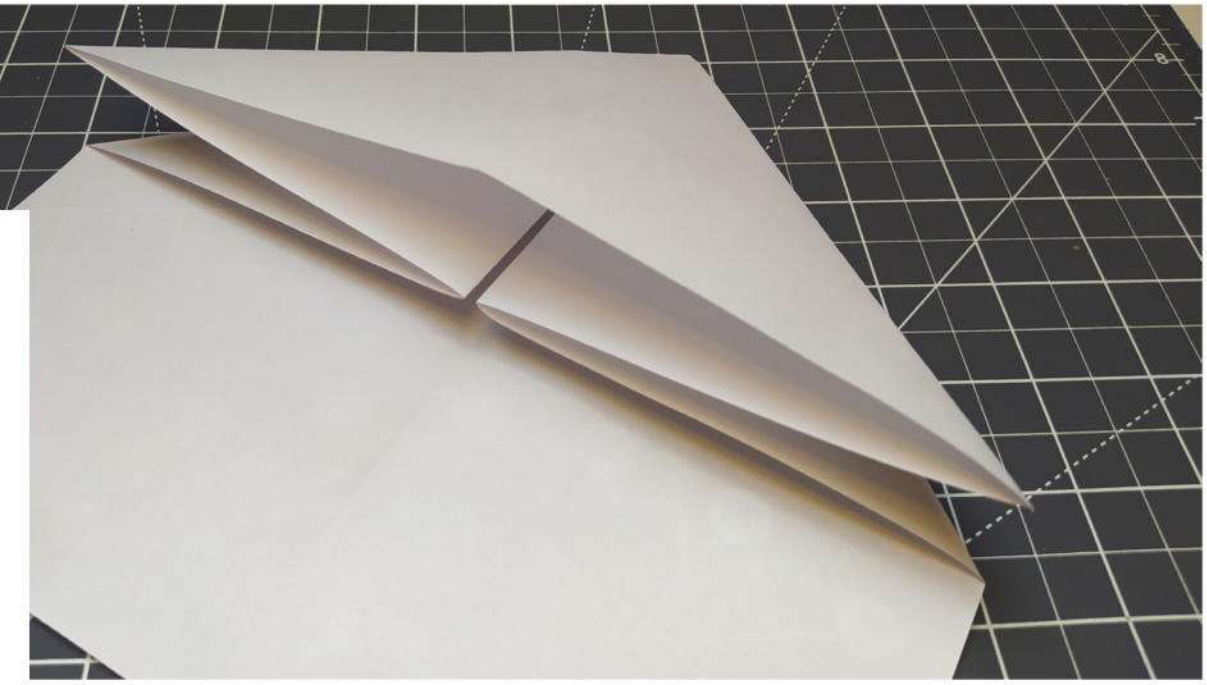


Figure 5
A complex fold creating a layered triangle

few pages of 'white wings' paper in the back for you to use to build these models. This was really useful as you could feel the weight of the stiff white wings paper – it actually feels more similar in weight to thin crafting cardstock.

One approach would be to print the pages to scale onto thin cardstock and cut them out, which would work excellently. However, as we have access to small vinyl/craft cutting machines, we decided to scan some of the plans, tweak and edit them, convert them into vector drawings, and then cut them using the vinyl cutter. While this takes more time for the first one than printing and cutting, it's useful if you want to make a few to play with or perhaps to give away to someone else.

> "These were a new type of paper aeroplane to me – **I was interested to see how well they flew**"

We tried numerous designs from the book, and the process was as follows. Use a flatbed scanner to get as clean a scan as you can from the page; try to keep the page in line with the scanner as much as possible so that vertical lines in the designs on the page stay parallel when scanned. Once you have an image, or scanned PDF of the design you want to try, import that image into the free and open-source Inkscape software.

With the image selected, click Path > Trace Bitmap. As our image should be pretty much black on white, usually the default setting of the Trace Bitmap tool will correctly trace the image and create vector paths for the strokes. Once you click 'Apply', you should have a vector copy of the scan overlaid on the original import. Separate the two images by moving the vector copy slightly and then delete the imported scan. What you will probably find is that you have vector lines that have formed the inside and the outside of the lines in the scanned document – we need to get rid of the inner lines so that we just have one cut line on the outside. To do this, with the vector image selected, we can click Path > Break Apart. Doing this, you should see that every line in the traced scan is now an individual item. We can then click to select the inner lines of each component and press the **DELETE** key. We also, at this point, deleted any extra bits we had scanned that we didn't need or want the craft cutter to try and cut: page numbers, component numbers, etc.

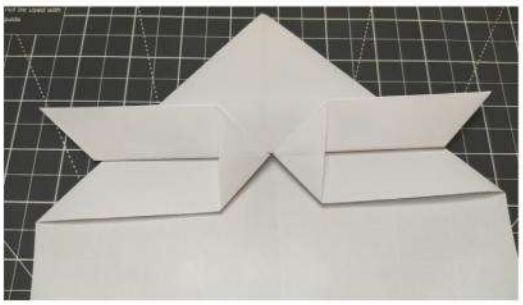


Figure 6
The hardest part of the build is done, with the landing gear sections created

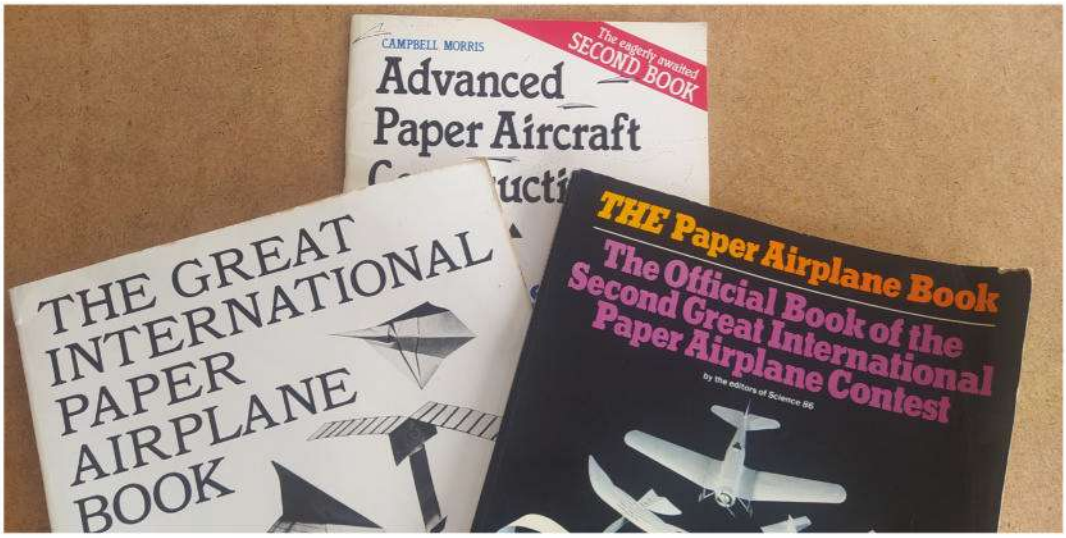


Figure 7
A small, but inspiring, collection of paper aeroplane books

You might find that, as you delete an internal line, your aeroplane components become filled as they become a solid object. It can be helpful sometimes to switch to View > Display Mode > Outline to see what is going on with your vector cut lines.

Once happy, we sent the job over to the cutter and cut the design out of thin cardstock. Almost all laminated paper aeroplane designs use some larger, full-length fuselage pieces and then add smaller sections at the front of the fuselage to increase the nose weight. Whilst the book suggests a range of glues you could use for this, we tended to just use a common glue stick.

We scanned, converted, and put together a few of the designs in the 1985 book, but the one we were most impressed with in terms of distance flying and stability was the 'Rainbow' by Hironori Kurisu, who was only ten years old at the time of the competition, you can see this aeroplane (yellow) in **Figure 8**.

Armed with some knowledge from the designs in the book, we designed our own laminated aeroplane – the Inkscape SVG files are available to download at **hsmag.cc/issue66.** You can see ours (green, centre) in **Figure 8**. Of course, you can either print and cut out the design or, if you have access to a vinyl cutter, you could automate your cutting. Our design is made up of three full-length fuselage sections laminated together; the centre section is a solid fuselage profile with a vertical stabiliser or tail fin. On either side of that main section are two full sections without a tail and with small fold-out tabs to receive the wing and the horizontal stabiliser section. Before fitting the two sections, you need to use a ruler and an object to make a scoreline with on the two tabs. A good option would be a thick darning-needle, but you could use something like a blunt cutlery knife to create these scores. First, consider which side of the section you need to score; we want to end up so that, when the three centre fuselage pieces are laminated together, they have the tabs sticking out on either side (**Figure 9**). Next, take the two medium-length fuselage components, which are around 60 mm long, and glue one on each side of the fuselage at the nose end. Don't add any of the very small nose sections yet as you only need to add as many as you need to get the aeroplane to balance correctly, which we will do later.

The wing is made from an upper and a lower section, but we need to add some scoring and marks to help us assemble this part. On both the upper

> "Armed with some knowledge from the designs in the book, **we designed our own laminated aeroplane**"

QUICK TIP

We covered the basics of using Inkscape with a vinyl cutter in issues 58 and 59.



Figure 8
Three laminated paper aeroplanes; the central one is our design, with the others taken from the book

and the lower wing parts, make some small marks at the centre points of the front and the back edges – this means you can then stick the support part with the large circular holes in (to reduce weight) to the centre of the larger wing. Make sure that you still have some centre marks on the upper side of the wing so that we can score it and align it to the fuselage. Before we mound the wing, though, we need to score each side to form some dihedral tips. Measure in 40 mm from each end of the wing, and then make a scoreline on the upper surface. Similarly, lightly score the centreline of the wing on the upper surface. Finally, glue the wing to the tabs on your fuselage sections, making sure it is centralised correctly. The horizontal stabiliser slides over the rear fuselage underneath the rear tabs – you can glue this into position now. Once everything is dry, gently lift the wings so that they rise up at around 10 degrees from the centreline, and turn up the dihedral tips another 20 degrees or so. With this done, you now want to try and get the aeroplane to have a centre of gravity about halfway across the width of the wing. To estimate the balance point, place a finger in each of the large circles under the wing and try to balance it. You should find that it is currently tail-heavy, so glue on a pair of the small nose pieces and recheck the balance. Depending on the cardstock and glue you have used, you may need to add a few more; most of ours have had at least two small sections added on each side.

Once your aeroplane is balanced in the centre of the circles, you can give it a test flight. Try to find a large indoor space: a sports hall is an excellent choice, or seek a sheltered outdoor space when the weather is calm. Holding the aeroplane under the wing, we find a medium-hard push into the air can give us a decent flight time.

Hopefully you've found this article interesting, and perhaps it's inspired you to make some simple, or not so simple, paper aeroplanes. It's amazing how much you can learn about flight with just a few pieces of paper!

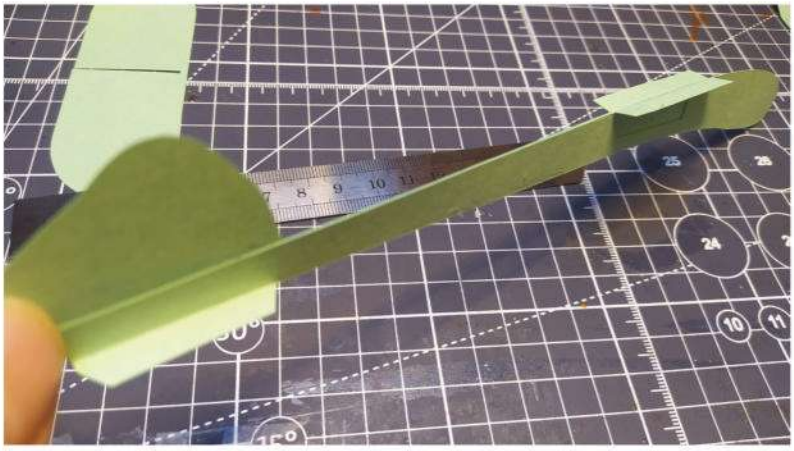
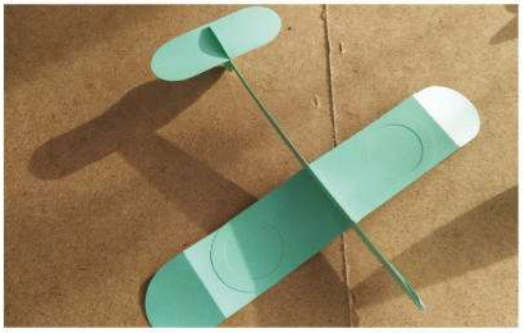
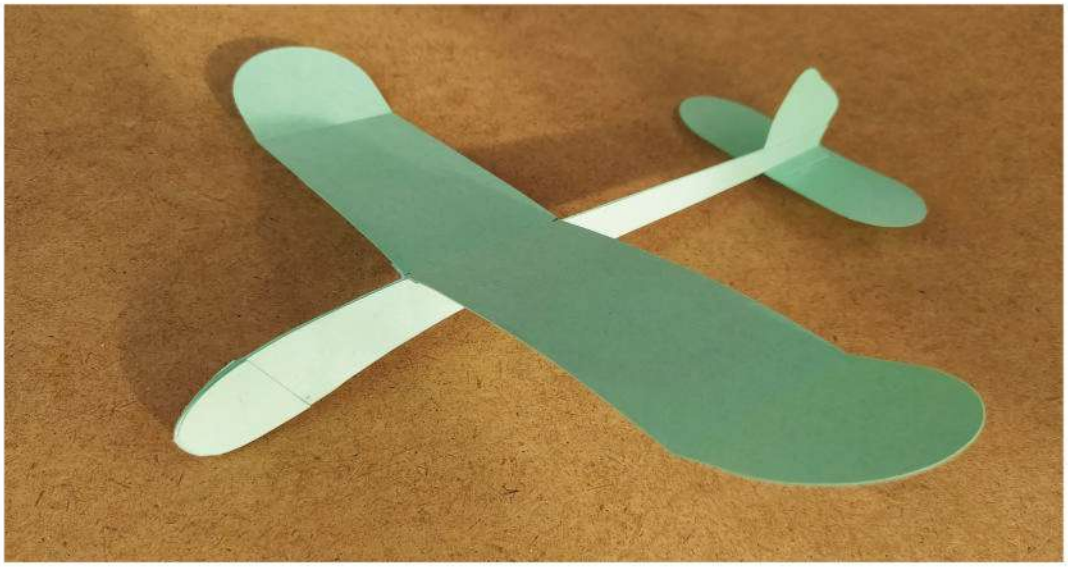


Figure 9
The main fuselage sections with folded tabs are glued together



Left
The underside of the laminated aeroplane design, showing the wing reinforcement section



Left
The laminated design, fully assembled

TUTORIAL



3D printing with nylon

When the going gets tough, print with polyamide



Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Above
Our first print came out looking really good

Nylon has two slightly separate reputations in the 3D printer world. Firstly, it's supposed to create high-quality parts, and secondly, it's supposed to be hard to print. Are either of these true? It is an 'engineering' filament? What does this even mean? Let's take a look inside the slightly confusing world of 3D printing nylon.

While most 3D printer filaments are unheard of outside the makersphere, nylon is a household name. Perhaps that's because it was one of the earliest plastics available – released in 1938, it was the first synthetic thermoplastic polymer. And perhaps it's because nylon fibres have found extensive use in the textiles industry. The word 'nylon' was once synonymous with stockings.

What is it about this clothing fibre that makes it so good for 3D printing? It has:

- A high-impact resistance and is exceptionally tough
- A low coefficient of friction, and good abrasion resistance
- Excellent layer adhesion, meaning it's strong in all directions.

Taken together, these three properties mean your 3D-printed parts are much more likely to survive use and abuse in the real world. Engineers have all sorts of numbers and equations for this sort of thing, but in more mundane terms, nylon prints are just more robust than any other common 3D-printed plastic.

Given that it's so great, why doesn't everyone just print in nylon? There are basically two problems: moisture and warping. Both of these are issues for other filaments as well, but they're very pronounced with nylon.

Nylon can absorb enough water to cause problems with just a few hours of exposure to the air. This is a problem because most prints take a few hours – and you may want to print more than one thing. Basically, you're going to need a setup for filament drying, and you're going to need to be able to print from

WHAT IS NYLON?

In this article, we've called the filament 'nylon', and that's what it's commonly called, but it's also sometimes referred to as polyamide (PA). There are slightly different formulations of nylon. The common ones you'll find as 3D printer filaments are PA12 and PA6.

PA12 absorbs less moisture than PA6. PA12 is also tougher than PA6 (thanks to its slightly higher flexibility), but not as strong (regarding load capacity).

Nylon filaments probably span a larger price range than almost any others. You can get it for under £30 per kilo, but then you can also spend a couple of hundred pounds on a kilo. At first glance, you might not notice much difference.

Cheaper nylon filaments may not even mention what type of nylon they are, and may be a bit harder to print with. That said, as long as you're not trying to push your print to its absolute limit, it might be sufficient.

More expensive filaments usually have specific features engineered into them, so it's worth taking a look across the range to see what has the features you're after. Many nylon filaments are available as samples, so you can test out if a particular filament works for your application.

it. In years gone by, this often meant either some expensive industrial setup or quite a bit of hacking, but these days you can get a filament dry box that you can print from for under £50. This is basically an enclosed spool holder with a heating element.

Remember that you need to keep your spools dry in storage and then keep them dry while printing. You can dry out a spool, should it become damp, but it's going to take some time.

The second big problem with nylon is warping. Like many plastics, nylon shrinks as it cools, and this means it bends, as some of the plastic will have cooled more than other areas. Depending on the part, the bend could be insignificant, enough to

> **Nylon shrinks as it cools, and this means it bends, as some** of the plastic will have cooled more than other areas

make it useless, or make it detach from the print bed mid-print. Minimising warping with nylon can come down to two things: temperature management (basically, this means keeping draughts away from your printer and, ideally, using an enclosure), and considering using a filament with an additive.

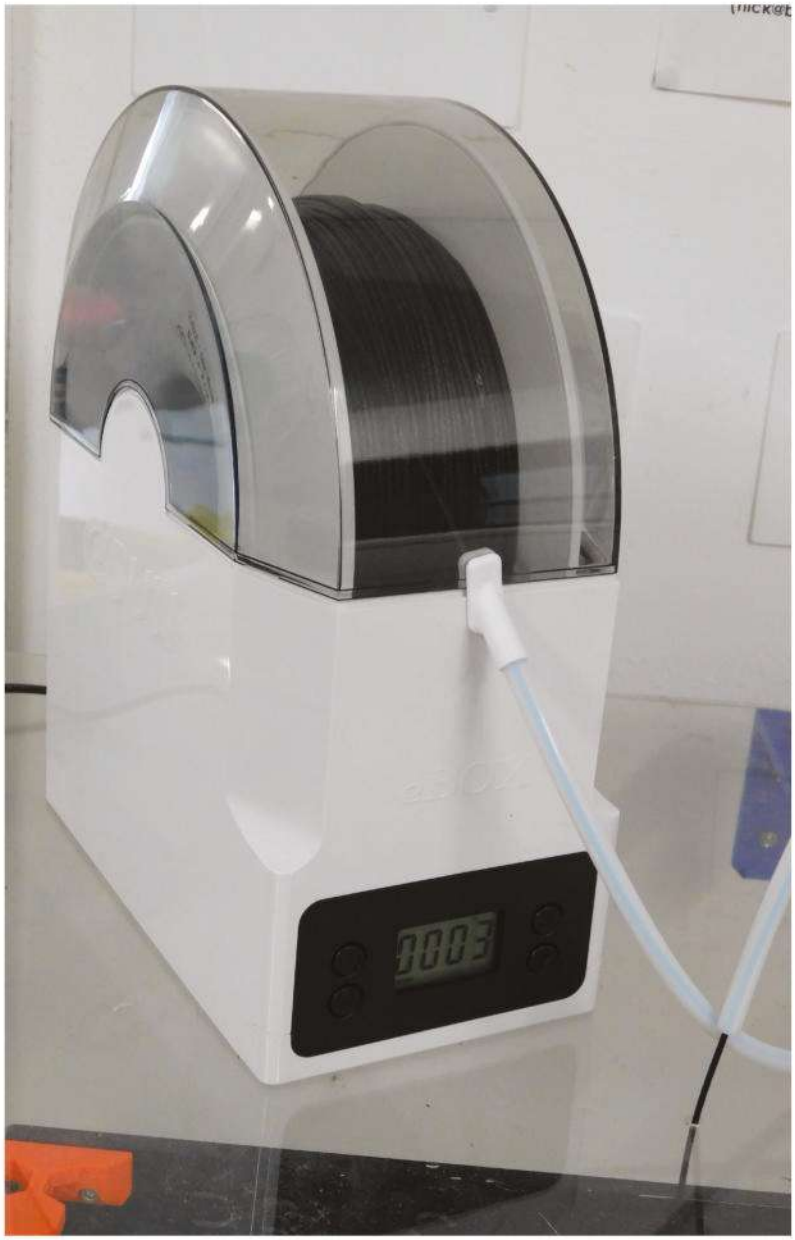
Carbon fibre (abbreviated as CF) is the most common additive to nylon, and it makes it stiffer and warp less. The amount of carbon fibre is usually given as a percentage, and can be as low as 5%, →



Below
Nylon comes almost exclusively in two colours: white and black

NYLON AS FLEX

Nylon isn't traditionally considered a flexible filament, but it is more flexible than most 3D-printable plastics. This little bit of flex means that nylon is a good choice for anything that needs a bit of bend, such as a living hinge or a snap-fit connector.



Below
A filament dryer that you can print straight from is pretty much essential for printing nylon

and goes up to around 30%. More carbon fibre will increase the stiffness of the material, but this in turn can reduce the impact strength, because the flex of nylon allows it to absorb force without breaking. Adding just a little carbon fibre to a part can make it easier to print without affecting the material properties too much, but you can get filament with more if you are looking for extra stiffness.

You can also find nylon reinforced with glass fibre, and even Kevlar. These have slightly different properties to carbon fibre-reinforced filament, and probably fit into the category of materials that, if you are doing work complex enough to need them, you probably already know more about them than an introductory magazine article will tell you, so we won't dig into them further.

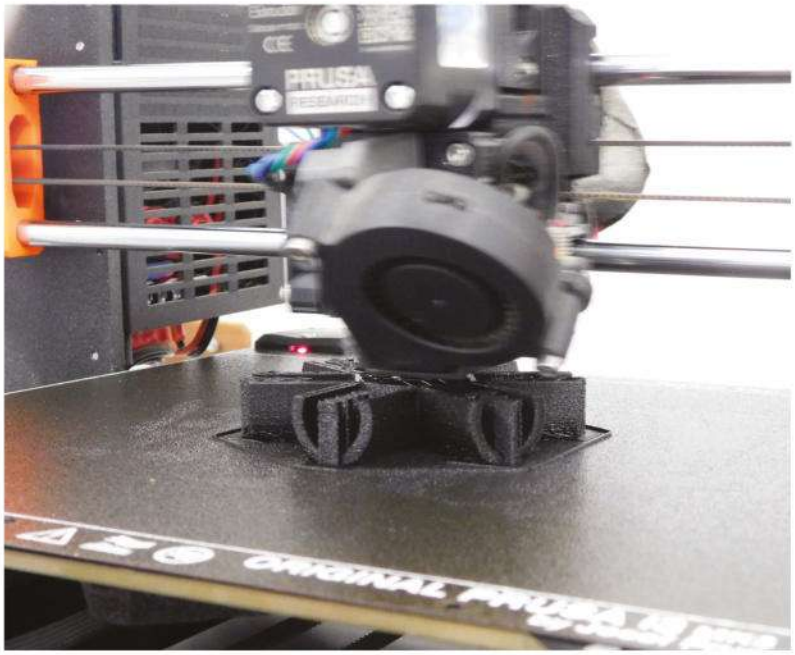
The main downside to carbon fibre is that it's an abrasive and can wear out your nozzle. Brass nozzles are too soft to print reliably with carbon fibre, so you'll need hardened steel, or one of the more exotic nozzles.

> **There's one final pitfall to overcome to** successfully print nylon – bed adhesion

If you've managed to deal with warping and moisture, there's one final pitfall to overcome to successfully print nylon – bed adhesion. Nylon isn't inherently bad at sticking to print surfaces, it's just that it sticks to different surfaces than most other 3D-printed plastics. We've done our testing, this issue, on a textured PEI sheet with glue stick

SINTERED NYLON

Nylon is now a staple of the extruded-plastic 3D printer world, but this isn't the only way of printing with nylon. SLS – or selective laser sintering – is a process where nylon powder is melted and joined by a laser. It's a common form of industrial 3D printing that hasn't really made it into the hobbyist world. While buying a nylon SLS is out of reach for most makers, it is possible to get parts made for a reasonable price. If you need high-quality nylon parts, and can't get them on a home printer, on-demand nylon SLS is an option.



Above
We printed on a textured PEI bed with glue stick, but specialised nylon print beds are available

applied. This works, but there are better options, and many manufacturers make sheets specifically for nylon.

That's all the bad news about how to print nylon; here's the good news: it prints really well. You don't need any particularly outlandish temperatures – if your printer can handle ABS, it can probably handle nylon. Once you've overcome the above, you'll probably find you can get really good prints. Overhangs, bridges, and fine details typically come out well.

The only issue we've had with nylon prints is that supports can be hard to remove. Nylon has excellent layer adhesion and this means supports tend to get stuck on. You have to be a little bit more careful about the design and slicing of your parts to ensure that this doesn't cause you problems.

Printing well with nylon isn't really difficult, it's just a matter of making sure you have the right equipment – and for many people, that's just a case of adding a dry box. If you don't already have an enclosure, this can be as simple as an upturned cardboard box. If you're printing parts that need to survive a bit of abuse, then nylon is an excellent material to have in your toolbox.

RECYCLING

3D printing uses a lot of plastic, and nylon is typically oil-derived. As responsible inhabitants of this planet, we should be minimising the amount of oil we use. If you need to use nylon, then you can still do your bit for the Earth by using recycled nylon if possible. This is available from a few sources.

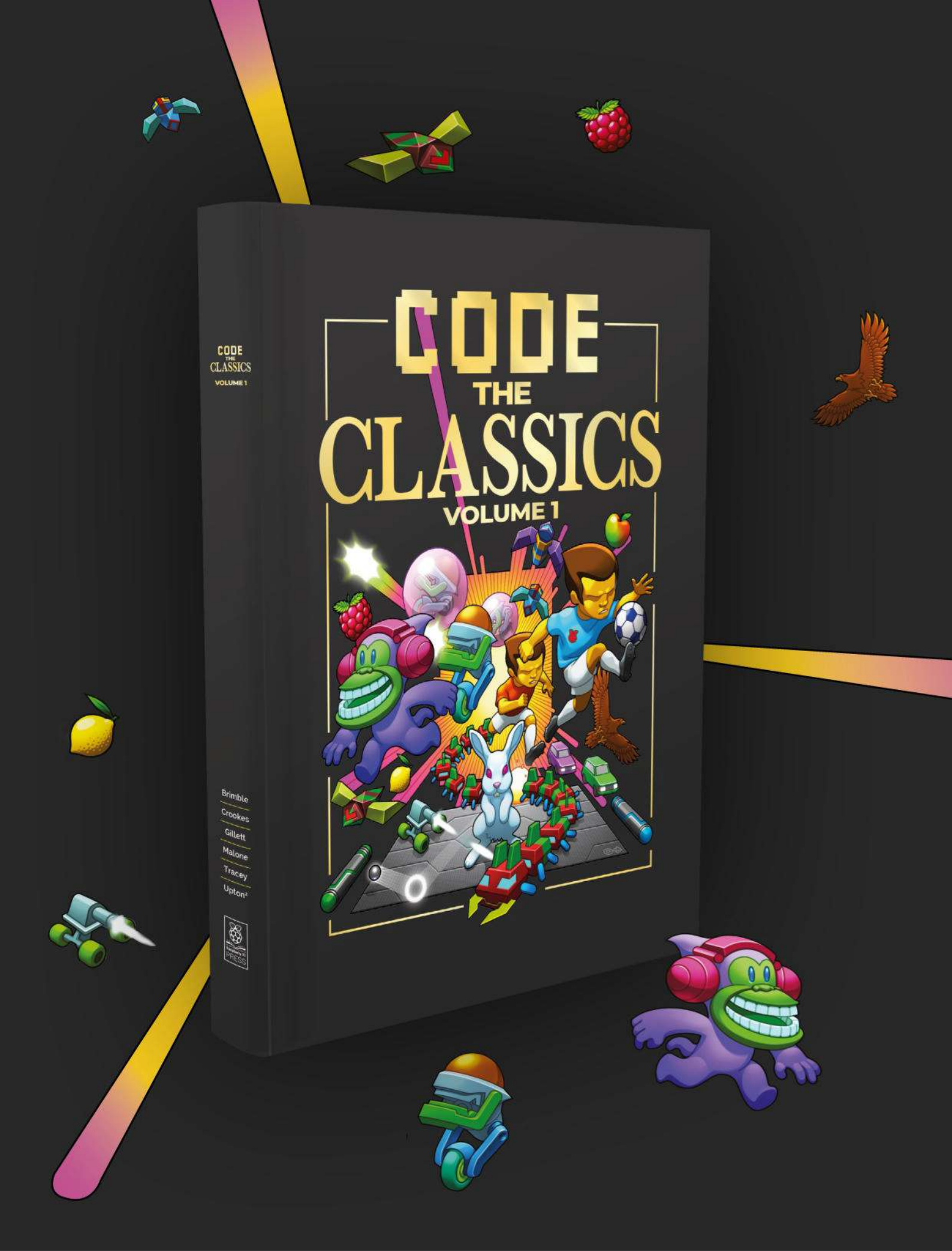
Fiberlogy's 'R' range of filament includes nylon. At the time of writing, Fishy Filaments – which creates nylon filament from recycled trawler nets – had a slight hiatus as it switched distributor. However, by the time you read this, the firm's Porthcurno and OrCA filaments should be available worldwide from Fillamentum. We hope to have some in the office for testing soon.



Above
Mechanical parts, such as gears, are a great use of nylon filament

CARBON FIBRE

It's common to find nylon filament with carbon fibre in it – this fibre is typically chopped into short lengths. This provides some stiffness and can be printed without any special equipment. However, you can get stronger parts by printing parts with a continuous thread of carbon fibres running through the filament. Be aware that this requires a specialist printer.


CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett
Malone
Tracey
Upton²
PRESS





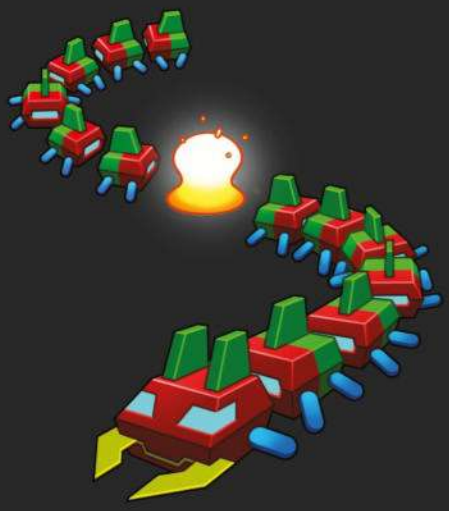
CODE THE CLASSICS VOLUME 1

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.





- ***Get game design tips and tricks from the masters***
- *Explore the code listing and find out how they work*
- ***Download and play game examples by Eben Upton***
- *Learn how to code your own games with Pygame Zero*



Available now **hsmag.cc/store**

TUTORIAL

Amstrad CPC expansion

Connecting Raspberry Pi Pico to a Z80



Ben Everard

@ben_everard

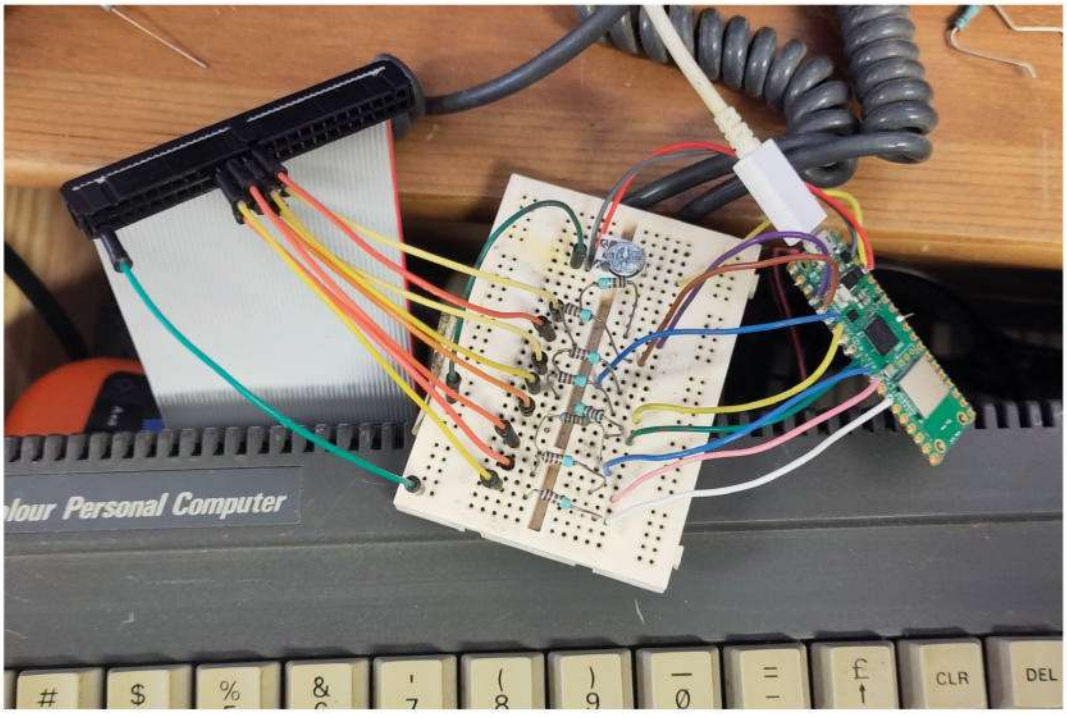
Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

During a recent refurb of Bristol Hackspace, we found that an Amstrad CPC that had been in a pile of computers assumed broken was, in fact, fully functional. Well, we think it's fully functional, but we don't have any software for it, or any way to load software at the moment. It does, however, boot to BASIC.

As there's nowhere to save data, there's not much point in creating complex programs because they'll be gone when you power off the computer. There are still a few sources of 3-inch (no, not 3.5-inch) floppy disks around, so we could get some, but to be honest, I've started enjoying this ephemeral style of computing. Turn the computer on, have a little experiment, see what happens, then turn it off and it's gone forever.

I quickly found that when trying to do something interesting in around 10–20 lines of BASIC (this seems to be the amount I don't mind losing when the electrons stop), I got the best returns using the sound command. There's a very particular bleepy-bloopy sound that's very nostalgic, and very easy to make on these machines.

Quite quickly, I started to wonder about adding some extra hardware. After all, there's an expansion port on the back of the CPC, so what would it take to plug in a bit more hardware? The first thing we wanted to make was an analogue input. This could



Right
We used a potentiometer to create a voltage input, but we hope to hook this up to a modular synth

be a sort of minimal test to see if it was easy to interface with the CPC and generally play with the technology involved.

The expansion header has a whole bunch of exposed pins on, only some of which we understand – we're still digging through the documentation.

We got a connector for the port and started exploring. The main set of pins we wanted to experiment with was the data pins. There are eight that let us send 8 bits at a time to the CPC. There was also an additional set of address pins, but we planned on ignoring those for the purposes of this experiment – as we only wanted to read from one address in our code, we could just send our 8 bits and ignore the address completely.

The pins were pulled up to about 3 V – we could pull them low to change them. This meant we could hook them up to a Raspberry Pi Pico without worrying about level shifters. We stuck a 10 kΩ resistor between Pico's pins and the CPC inputs for added safety. We've not found much documentation about exactly how these read data, or best practices for hooking things up to them, but so far, this seems to work.

With this super-simple circuit in place (the ground is connected, and the eight data pins are connected to eight Pico GPIO pins via 10 kΩ resistors), we were able to write data out of Pico and read it into the CPC in BASIC using the following BASIC command:

```
print INP(&FBD0)
```

The **&FDB0** doesn't do anything – that tells the CPC what port to read, but since we're ignoring the address anyway, this will just read whatever we're shoving into the data bus.

TALKIN' 'BOUT SOUND GENERATION

Now we had a way of connecting the two, what should we do with it? As we'd already experimented with sound, we decided to hook this into a simple sound generator and create a program that makes a sound based on a voltage sent into one of the Pico's analogue inputs.

All we needed to do was convert the analogue signal into an 8-bit output and then push this to the GPIO pins. The following MicroPython code does just that:

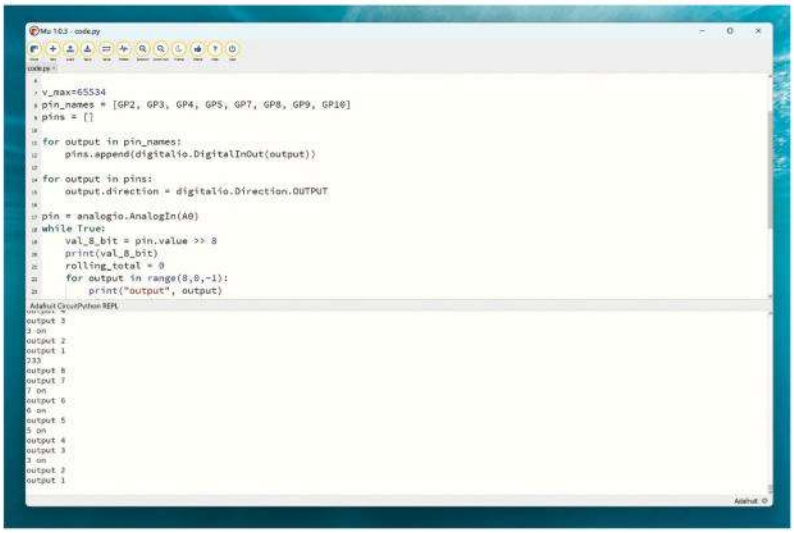
```
import analogio
from board import *
import time
import digitalio
from math import pow

v_max=65534
pin_names = [GP2, GP3, GP4, GP5, GP7, GP8, GP9, GP10]
pins = []

for output in pin_names:
    pins.append(digitalio.DigitalInOut(output))

for output in pins:
    output.direction = digitalio.Direction.OUTPUT

pin = analogio.AnalogIn(A0)
while True:
    val_8_bit = pin.value >> 8
    print(val_8_bit)
    rolling_total = 0
    for output in range(8,0,-1):
        print("output", output)
        size = pow(2,output)
        if val_8_bit > size:
            pins[output].value = 1
            val_8_bit -= size
            print(output, "on")
        else:
            pins[output-1].value = 0
    time.sleep(0.1)
```

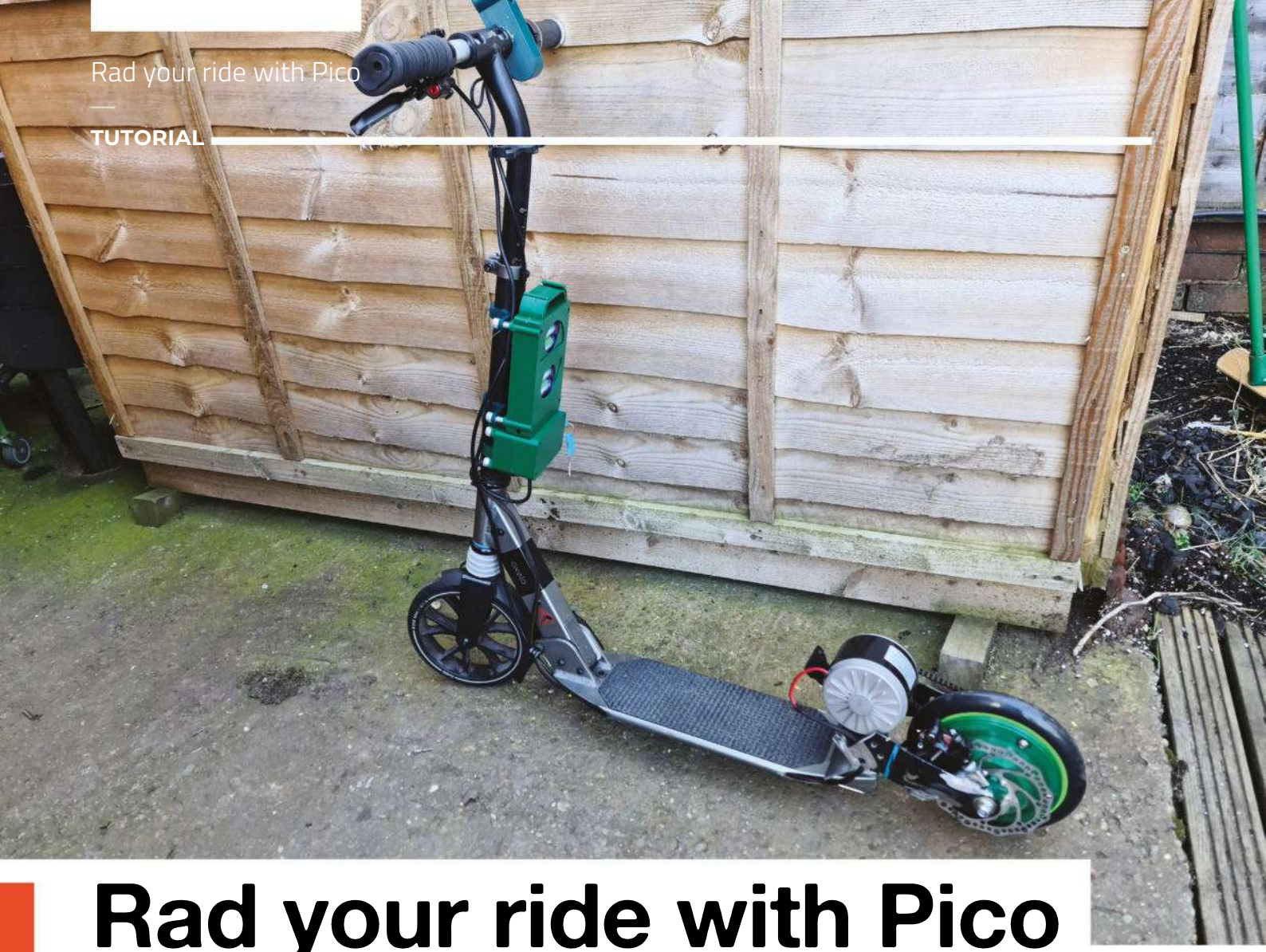


Above
The code gives you details of which IOs are switched on and off

Finally, we just needed a bit of BASIC code for the CPC to turn this into annoying beeps and bloops:

```
10 SOUND 4,10*INP(&FBD0),1,5,0,0,0
20 GOTO 10
```

This obviously doesn't take into account any musical theory or keys. It just turns a voltage into an 8-bit beep. We're going to keep experimenting over the next few months to see how we can progress this connection into a more useful setup.



Rad your ride with Pico

Turn a manual scooter into a motorised monster



Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

Scooters are fun, and can be a good alternative to cars and bikes when considering the daily commute. They can even be useful while at work if your job involves moving between campus buildings several times a day, or travelling back and forth in large warehouses or hangars. Electric scooters are even more fun, but they are also expensive. In this project, you'll see how to take an ordinary manual scooter and make it electric without resorting to expensive hub motors or conversion kits. This is a two-part project. In the first part, we'll take a look at the components we'll use and how we'll need to modify our scooter to make it all work. In the second part of the project, you'll see how to power your motor from custom batteries.

The anatomy and function of an electric scooter isn't mysterious or difficult to understand. The user stands on a platform, activates a throttle, a motor hums into life, and the wheels of the scooter begin to turn. When the user wants to stop, they release the throttle and apply a brake or use their foot to slow down.

A manual scooter works in a similar way, but with more leg action and less throttle – surely we just need to add a motor, some batteries, and a speed controller, and that's everything, right? Well, it isn't quite that simple. Firstly, manual scooters come with very primitive brakes (if they have brakes at all). The wheels on the scooter are typically not designed to be driven, and just have a simple bearing and possibly a fork with a suspension spring. Making a manual scooter move under motor power means making some very permanent changes to the chassis of the scooter, mounting the motor near to the rear wheel, enlarging the rear fork to accommodate better brakes and a drive sprocket on the rear wheel, and modifying the

wheel itself so that it can handle the extra stress coming from the motor.

Most of the issues with the wheel can be dealt with by using some 3D-printed parts and a healthy dose of epoxy resin to reinforce the wheel. The drive sprocket for the motor and the disc to upgrade the brakes can be mounted on opposite sides of the rear wheel using 3D-printed parts, and using

> **Once you have a solid wheel, you can build your** rear fork to match its width

plenty of resin to fill the space between the two sides will make sure that the bolts have plenty of solid wheel material to make contact with when the forces from the motor and brakes are transferred through them.

Once you have a solid wheel, you can build your rear fork to match its width. If you have access to a welder, you can do this very quickly by cutting the original rear fork into pieces and re-welding it with →

OFF-ROAD

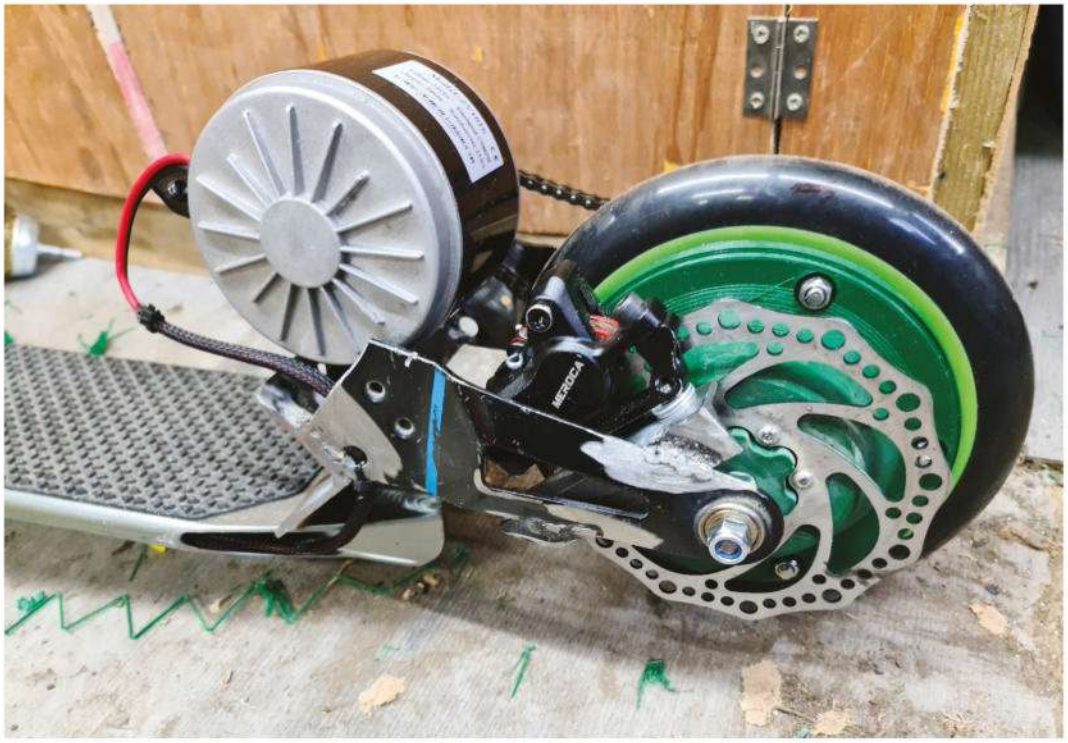
If you're thinking of using an electric scooter in the UK, you're going to be disappointed to learn that you're not going to be able to ride it on the open road, or on a footpath, or even in a bike lane. Electric scooters are in a class of vehicle known as 'powered transporters'. It's the same place you'll find one-wheels, hovercrafts, and electric skateboards. In UK law, a powered transporter is the same as a motor vehicle, and as such you would be expected to license and maintain it in exactly the same way as a car or motorbike. Unfortunately, scooters don't have some of the important features needed to pass an MOT (Ministry of Transport) test (like headlights, indicators, registration plates, etc.) – and so they will not be legal to ride on the road, and you can't get insurance to drive them. However, they are still technically classed as motor vehicles, so you aren't allowed to ride them on the pavement either. If you *do* ride on the pavement or on the road, you could be fined several hundred pounds and risk losing your driving licence. You can ride them on private land – such as corporate campuses – with the landowner's permission. At the time this article was written, the government is developing new standards and adjustments that will eventually make electronic scooters road-legal in the UK. How long that process will take depends on how quickly the government decides to move.

TAKE CARE

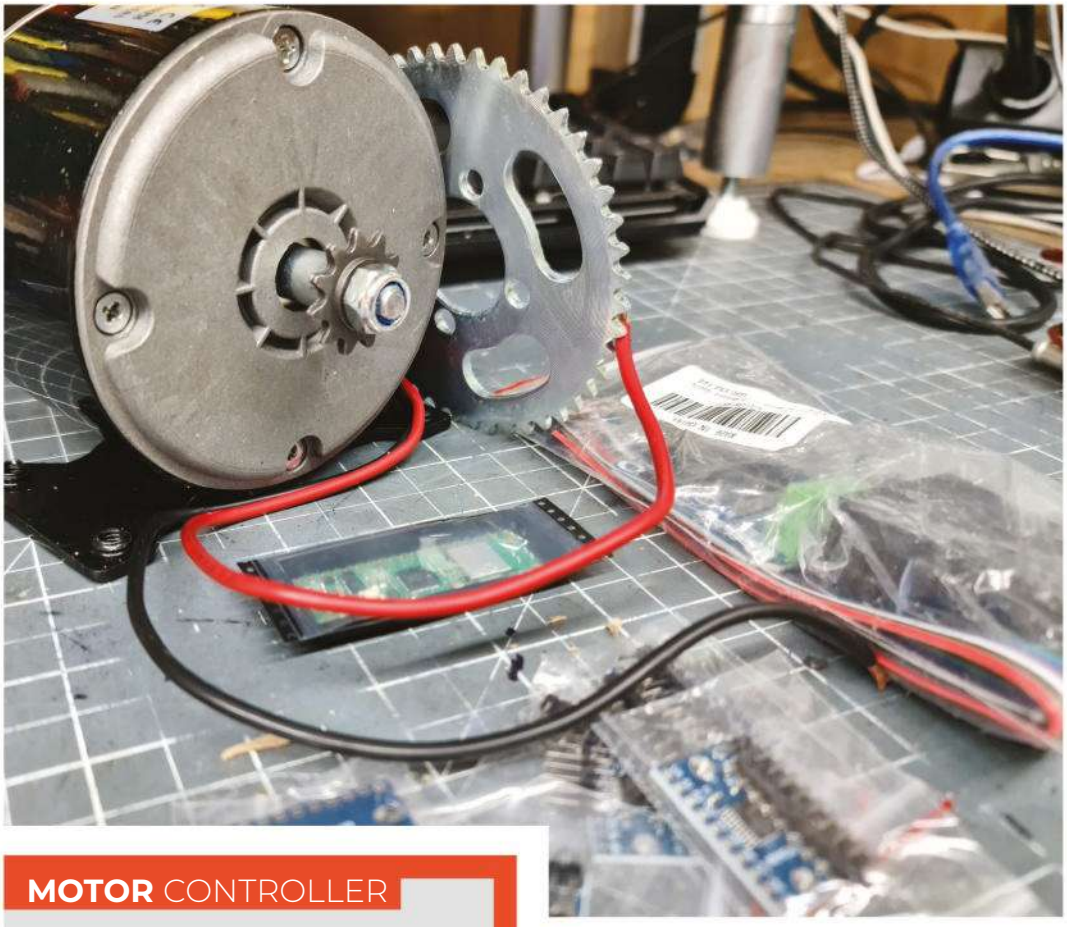
Scooters – whether powered or not – carry some risks. It's your responsibility to make sure the scooter doesn't go faster than you personally feel comfortable with. You are also responsible for ensuring that any chassis modifications you make are sufficiently strong. We strongly recommend you err on the side of caution as your decisions on both these risks could impact not just you, but the people around you.

QUICK TIP

Cycle chains can be joined with a split link, so you don't have to buy a chain exactly the right size. Simply remove excess links, join the chain with the split link, then tension the chain by moving the motor.



Left
Most scooters don't really bother with an effective brake, because they don't go very fast and you can stop them with your foot. If you're on a motorised scooter, you'll need a more powerful rear brake to arrest your more considerable forward momentum. A disc brake from a bicycle will do the job. Just remember that the brake will only stop the scooter, not you



Right
Using a motor with a sprocket and chain is much cheaper than using a hub motor and allows you to increase the torque from the motor mechanically

MOTOR CONTROLLER

Most BTS7960 motor controllers are H-bridge controllers that have a similar pinout. The boards have two power inputs, one of which (often labelled as pins B+ and B-) supplies power for the motor. Next to this input are the connections (M+ and M-) that directly attach to the motor.

The other power input is used to supply the logic side of the board, and is labelled as Vcc and GND. This logic level power input runs at 3.3V or 5V and is isolated from the motor power connection for safety. With a Raspberry Pi Pico, you should be powering this with 3.3V. The remaining connections on the board are used to control the speed and direction of the motor, and optionally monitor the current used by the motor. We won't be using the current-sensing features, so the pins labelled R_IS and L_IS can be ignored. The L_EN and R_EN pins enable the motor to turn clockwise or anticlockwise when they are high, or disable the motor when low. Obviously, only one of these should be enabled at a time. The LPWM and RPWM pins accept a PWM signal from the Pico, or a solid logic high to power the motor at 100% power. Since a scooter generally doesn't need a reverse gear, you only need to connect either the left or right enable and PWM pins to the Pico, depending on which way around your motor is connected. Assuming that you have a right-hand motor setup, you would make the motor move by setting the R_EN pin HIGH and providing a PWM signal to RPWM.

> **You will probably find that you won't want to run your** motor at 100% speed under any circumstances

a larger gap. If you don't have a welder, you can achieve similar results using threaded bar and locking nuts with a spacer between each side of the fork.

For something like a light scooter, a 20 amp, 24 volt motor in combination with the gear reduction from the sprocket drive should have plenty of power to move an adult at terrifying speeds. By controlling the power to the motor using PWM signals from a Raspberry Pi Pico, you can provide very smooth speed control without losing as much torque as you would if you simply reduced the overall supply voltage. You will probably find that you won't want to run your motor at 100% speed under any circumstances, and will throttle the maximum PWM value to something less than bowel-voidingly high speeds.

QUICK TIP

In MicroPython, the Pi Pico PWM output ranges between 0 and 65025, so setting a PWM pin with `pwm.duty_u16(32512)` would give you about 50% power.

ANALOGUE IN

Setting the motor speed is only half of the process; you also need to get a value from the user by reading a throttle control. It's tempting to think about creating a simple throttle control using a potentiometer, but that would be a bad idea. If you've ever used an old radio, you'll know that the potentiometer used in the volume control gets worn out, and the volume of the radio will suddenly change or crackle at full volume for a fraction of a second. This is annoying in a radio, but if it happens on a motor controller and the motor suddenly speeds up to 100% without warning, the result could be catastrophic. To avoid this problem, most electronic throttle controls use a Hall probe and a magnet to ensure smooth speed changes. The Hall probe and magnet aren't in physical contact and there are no tracks to get worn, dirty, or broken. The probe normally has two power pins and a signal pin, which outputs an analogue value to reflect the strength of the magnetic field it detects. As the magnet moves closer, the field gets stronger. Checking the position of the throttle controller is as simple as connecting up and polling the analogue pin of a microcontroller, just as you would with a potentiometer.



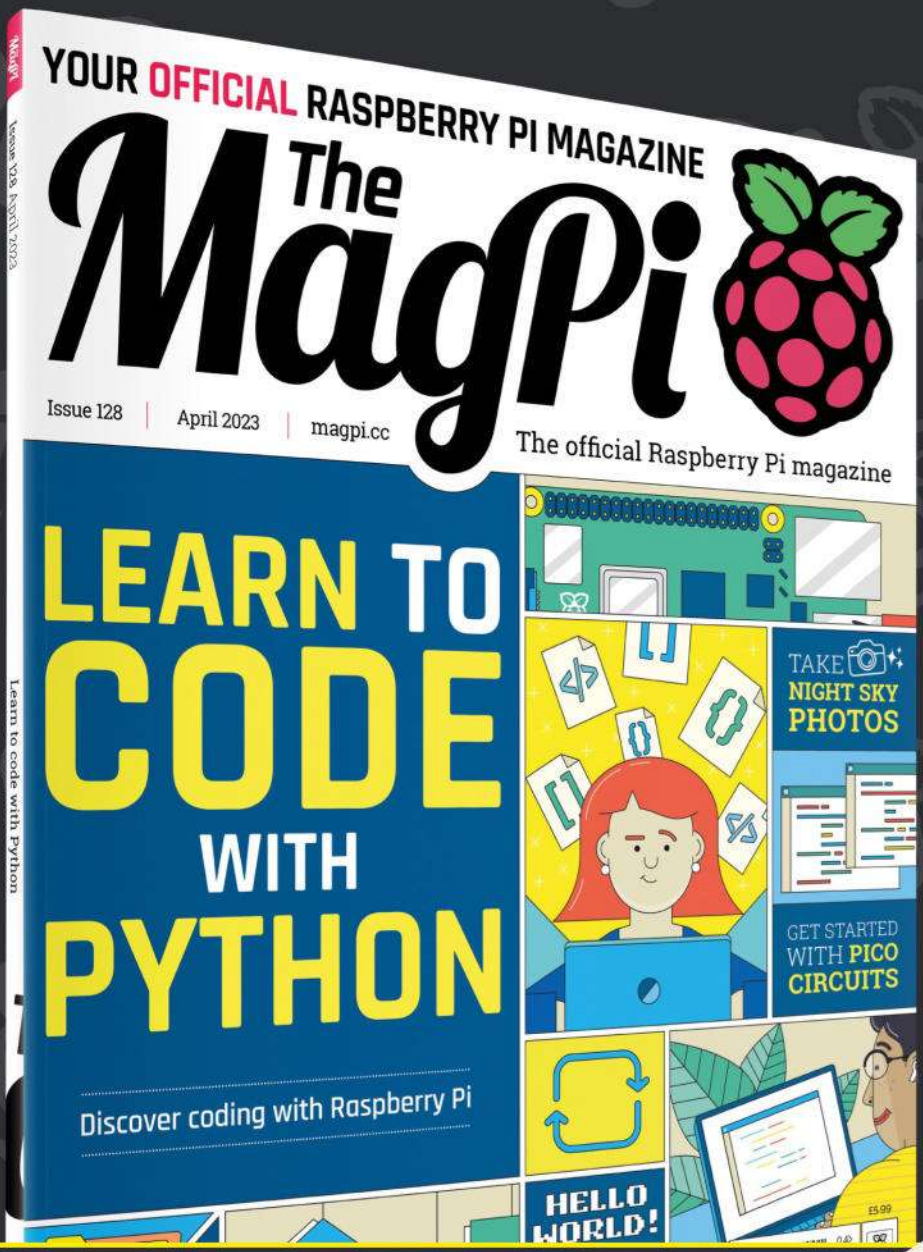
Above
You will need to mount your motor so that the motor's sprocket matches up with the sprocket on the wheel. The easiest way to do this is to fix a mount for the motor on the rear of the fork with slotted holes, and use the adjustment that the slots provide to position the motor correctly. You can also use this technique to tension the chain between the motor and the wheel



Left
You can see here how much wider the wheel becomes once you allow for the sprocket and brake disc. Widening that rear fork might seem a bit daunting, but it's actually pretty easy to do. Getting rid of the in-built suspension makes things much easier if you're struggling, and it won't actually make that much difference to the feel of the scooter

Above
To control a large motor, you will need to use a powerful motor controller like the BTS7960, which is often found on motor control modules that can handle around 43 amps, when appropriately cooled

DON'T MISS THE **BRAND NEW** ISSUE!



SUBSCRIBE FOR JUST £10!

- **THREE!** issues of The MagPi
- **FREE!** Raspberry Pi Pico W
- **FREE!** delivery to your door



+ FREE RASPBERRY PI PICO W*

Three issues and free Pico W for £10 is a UK-only offer. Free Pico W is included with a 12-month subscription in USA, Europe and Rest of World. Not included with renewals. Offer subject to change or withdrawal at any time.

magpi.cc/subscribe

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG 92
KARAKURI
Turn paper into machines



PG 94
ARDUINO GIGA
The most powerful Arduino ever

PG 96
POCKET REFORM
The tiniest open-source laptop





PG 86
BEST OF BREED
Things that are actually in stock (for now)

ONLY THE BEST

Yes, you can buy these products - they are in stock…for now!

Products that, at the time of writing, were in stock

By **Marc de Vinck** @devinck



Stock levels. It's been an issue for all of us for a while now. Cars, appliances, and anything that has some kind of electronics, especially microcontrollers, seem to have experienced a supply chain issue at some point over the last few years. But, then again, even paper towels or general commodities have been hard to come by at times. It's been a rough few years trying to acquire parts for your projects, but that seems to be getting a bit better.

All is not lost, especially if you are looking for microcontrollers, rather than a single-board computer. Raspberry Pi started selling the RP2040 integrated circuits to manufacturers so they can continue to support, develop, and innovate in that ecosystem, and it's been great! And the most important spec: they are in stock!

In this roundup, we'll be looking at a few new boards and products, but also featuring a few that we have seen before. The key theme being that, at least at the time of these reviews being written, they are all in stock, and they all look like they should continue to be in stock. So, if you have plans for making a cool new IoT device, robot, or so many other DIY electronic projects, don't worry about stock levels: there are some options out there.

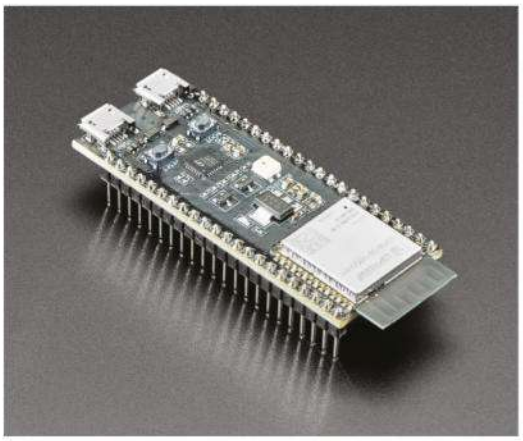
And yes, you may not be able to do all the things you imagined, but you can still do a lot, and you can still have fun!

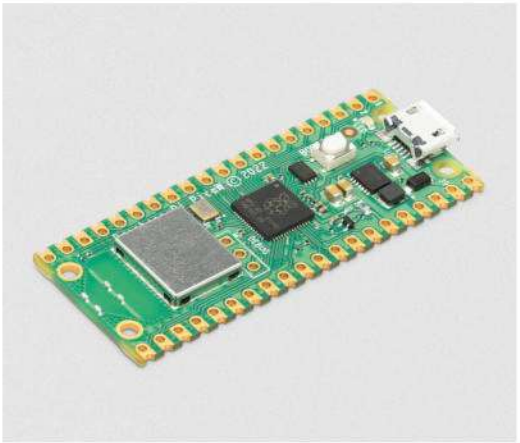
ESP32-S3-DevKitC-1-N8R8 vs Raspberry Pi Pico W

ADAFRUIT $19.95 | adafruit.com **PIMORONI** $6.46 | pimoroni.com

The ESP32-S3-DevKitC-1, designed by Espressif Systems and available from Adafruit, is an entry-level development board for the ESP32-S3-WROOM-1. It features an ESP32-S3, an Xtensa 32-bit LX7 CPU that operates at up to 240MHz, Wi-Fi, and Bluetooth LE with an on-board PCB antenna, and 8MB flash memory and 8MB PSRAM.

That's a lot of power and memory! There are a few variations of the board that are available, so be sure to check out the link to the product page. Just keep in mind, programming these types of integrated circuits can be more difficult than something like an Arduino, but you also typically get a lot more features.





How can you go wrong with the Raspberry Pi Pico? It's the original microcontroller based on the RP2040 and made by Raspberry Pi Ltd itself. Not only is it very affordable, but it also features a dual-core Arm Cortex-M0+ processor with 264kB internal RAM. You can do a lot with that combo! Since its launch, we have seen a lot of boards featuring the RP2040, but the classic Pico W with built-in Wi-Fi is still an excellent choice, especially if you are just getting started. And if you don't want to solder anything, be sure to check out the Pico WH variation with its header pins already soldered on, for just about a dollar more. It's great value!

VERDICT

ESP32-S3-DevKitC-1-N8R8

Powerful and in stock!

8/10

Raspberry Pi Pico W

The OG RP2040.

10/10

Yes, you can buy these products - they are in stock...for now!

BEST OF BREED

Badger 2040 W

PIMORONI $27.68 | pimoroni.com

A programmable badge with fast updating e-ink display! What's not to love? The Badger 2040 W features a Raspberry Pi Pico W and couples it with a beautiful and simple e-ink display. Don't think this is just a conference badge: it's an RP2040 with e-ink. You can use it in so many projects. Ignore the optional lanyard! But sure, it makes for a great badge too!

On the front of the Badger 2040 is a black and white 2.9″ e-paper screen, five buttons, and a slot to clip it onto a lanyard. On the back of the board there is a battery breakout connector, a reset button, and a convenient Qwiic/STEMMA QT connector, making the board very versatile for other projects. Conference or not, this is a great Pico W-powered board.



VERDICT

Badger 2040 W

Another fun 2040 board.

9/10

RASPBERRY PI DEBUG PROBE

PIMORONI $12.61

pimoroni.com

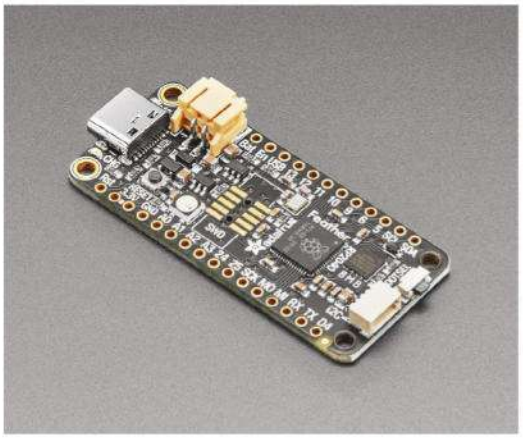
Got a Raspberry Pi Pico? Check out the Raspberry Pi Debug Probe by the creators at Raspberry Pi, and available at Pimoroni. The Debug Probe allows you to easily hook Pico's debug pins up to any computer via USB for debugging. It's also very handy to have for troubleshooting other microcontrollers and microprocessors. You can use it with either the processor serial debug interface or industry-standard UART. Both interfaces use a 3-pin debug connector and make debugging affordable and accessible. Check out the website for more information on this powerful little device.



Adafruit Feather RP2040

ADAFRUIT | **$11.95** | **adafruit.com**

The Feather ecosystem continues to grow, and the Adafruit Feather RP2040 still stands strong amongst all the variations of Feather boards. The board features an RP2040 32-bit Cortex-M0+ dual core running at ~125MHz, perfectly packaged into the Feather form factor. This allows you to add dozens of different shields and accessories with ease. Or, you can just design your own circuits and use up the 21 GPIO pins that feature four 12-bit ADCs, two I2C, two SPI, and two UART peripherals. It also has 16 PWM outputs that you can use for servos or LEDs. It's an impressive board, and a great choice for most projects.



VERDICT

Adafruit Feather RP2040

A great addition to the Feather ecosystem.

10/10

Raspberry Pi 3A+

RASPBERRY PI | **£28.50** | **raspberrypi.com**

So, how about any single-board computers that are in stock and ready to ship? Not all variants of Raspberry Pi are easy to get at the moment (though hopefully the situation will be improving by the time you read this), but the 3A+ is usually the most available. It's smaller than the model B, and it doesn't have the latest processor. However, for many embedded options, it's a great choice. The smaller size and lower power requirements mean it's more portable, and it's still got enough processing grunt for a huge range of tasks than microcontrollers struggle with. While it seems a bit unusual to recommend a board that's not the latest model, this really is a capable little computer.



VERDICT

Raspberry Pi 3A+

Powerful and readily available single-board computer.

10/10

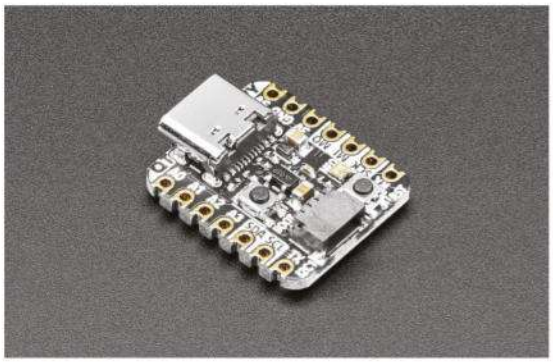
Yes, you can buy these products - they are in stock...for now!

BEST OF BREED

Adafruit QT Py RP2040

ADAFRUIT $9.95 | adafruit.com

After something a little smaller? The Adafruit QT Py RP2040 is ridiculously small and amazingly powerful too! The board features the popular RP2040 microcontroller and couples it with the growing ecosystem of STEMMA QT-compatible accessories. You get a reset and bootloader button, an RGB NeoPixel LED, and 13 GPIO pins. You can do a lot with this board. If your project needs to be small, yet powerful, check out the Adafruit QT Py RP2040.



VERDICT

Adafruit QT Py RP2040

You want smol, we got smol.

9/10

INKY IMPRESSION 7.3″

PIMORONI $71

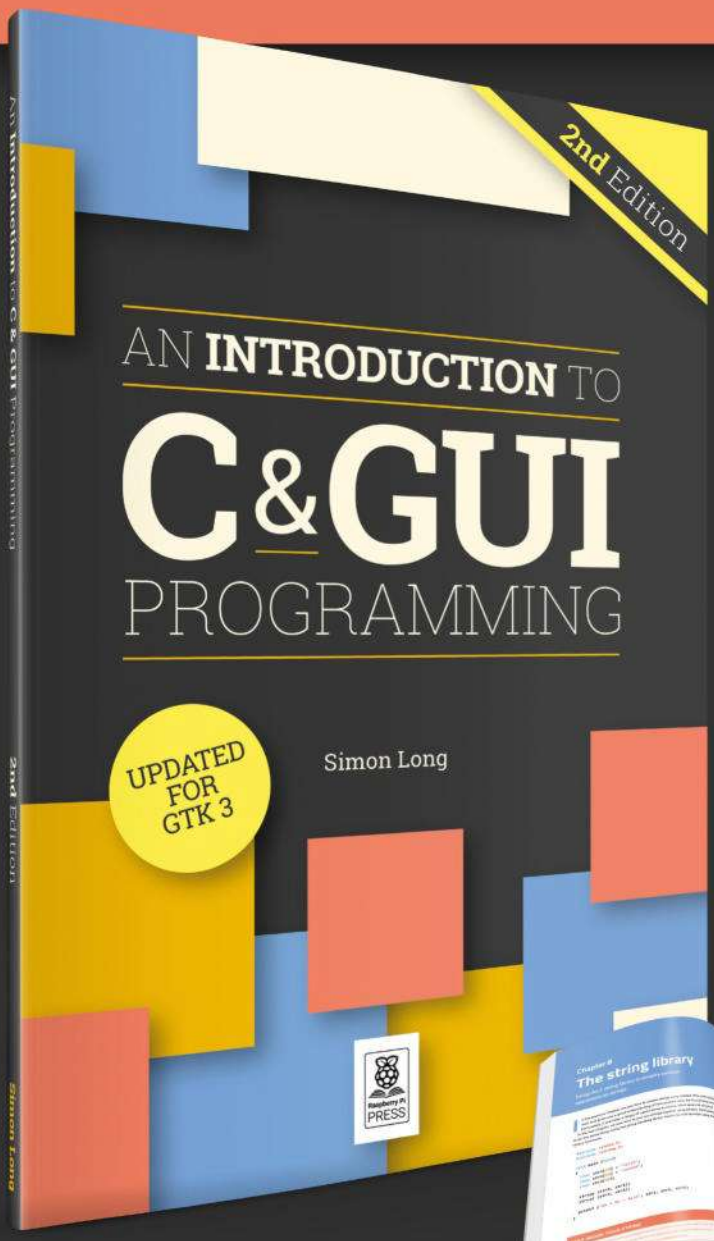
pimoroni.com

Admittedly I took a little liberty when including this in the 'it's available', as the Inky Impression from Pimoroni is available for pre-order and not in stock just yet. Although that will change once this is published, and I hope it stays in stock. I had to include this new display because of just how amazing it looks. No, it's not super-high definition, but it does have high pixel density, especially for a low power e-ink display. It's also fairly large at 7.3″, which is great for e-ink. But the real reason I had to include it in the roundup is because it's also a seven-colour display. Most e-ink displays are two or three colours – this one really shines at seven.

If you have a Raspberry Pi, this should be on your list of accessories.



AN INTRODUCTION TO C&GUI PROGRAMMING



All you need to know to write simple programs in C and start creating GUIs

Inside:

- Create simple command-line C programs
- Control flow with conditions and loops
- Handle variables, strings, and files
- Design graphical user interface applications in C
- Handle user input with buttons and menus
- Use advanced UI features such as data stores and dialogs

£10 with FREE worldwide delivery



Buy online: magpi.cc/cgui

Karakuri: How to Make Mechanical Paper Models That Move

Turning flat paper into 3D models

KEISUKE SAKA ◆ £9.39

By Ben Everard @ben_everard

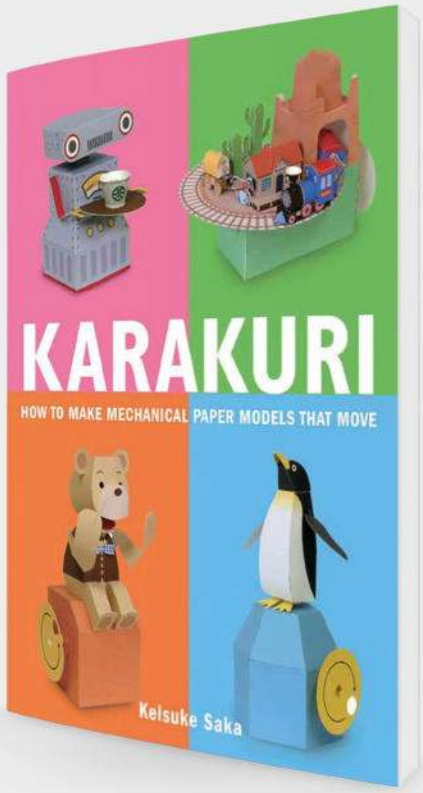
Paper is an almost magical material. In the tutorials section, we're looking at a childhood favourite – paper planes – and here we've been making some far more complex models, thanks to this book. This isn't a particularly new book, but is one that we've only just come across, so we're going to include it in this month's reviews section for all the other people who also haven't come across it yet.

Karakuri is a Japanese word for mechanised puppet – not unlike the Automata we've looked at in this issue's cover feature. In this book, Keisuke Saka shows you how to make them out of paper. You might hear the words Japan and paper, and assume that this is a book about origami but, while there are some similarities, the techniques Saka uses are different – most notably, because they involve scissors and glue. Well, we say scissors; you'll actually get a much cleaner cut using a craft knife.

As well as a look at the culture around these paper creations, the book contains twelve models to make. Eight of these are a look at specific mechanisms – gears, cams, etc. They're printed on white paper so that you can build on them to make your own designs. The other four are full-colour models – the ones that are on the cover: a penguin, a bear, a robot, and a train. They're fun to make and surprisingly addictive to use. This author has had one on his desk for most of this month while putting the issue together, and he constantly finds himself idly turning the handle while waiting for a meeting to start, or while reading an email.

CUT AND FOLD

The models are well explained and it's easy to follow the instructions, but you will need a steady hand and a very sharp knife to make them. They're all printed on cardstock, rather than the sort of thin paper you'd typically find in a book. This chunkier source material makes for sturdy builds (at least by paper standards), so that they survive a bit of use. The builds are quite involved and you do have to take your time. There's a particular type of making that means you lose yourself in the process. It's not necessarily hard, but requires concentration and careful movement over a long time. This author finds it a great option if he's had a bad day, or just wants to get out of a particular





Above
It's hard to convey just how delightful this penguin is

head space – it's almost meditative. Making these paper models is just like that. It's absorbing without being taxing.

A slight disappointment with this book is that you destroy the builds by making them. This is, of course, fairly obvious, but particularly for the 'white paper' mechanisms – which you can build on to make your own projects – it'd be great if there were a way to use them more than once – a downloadable template, for example. Or even the ability to buy just particular models without getting the whole book.

It feels a bit like you learn about making the models as you go through the book, but then get stuck when you get to the end because, while there's quite a bit about using the mechanisms, there's not much about designing the mechanisms yourself: how to make sure cams fit properly on the shaft, etc. It feels a bit like you're falling off the edge of a cliff when you finish the book and want to start out on your own.

We shouldn't focus too much on this, because we did have a lot of fun going through the book. There are a lot of models to work through before you get to the end of the book. As well as learning to work with paper, it gives you a chance to explore some mechanical concepts – the same sort that we've looked at in this issue's cover feature.

> "There are few areas of making where you can **make so much stuff with so little equipment**"

PRICE IS A FEATURE

There are few areas of making where you can make so much stuff with so little equipment. On the off-chance you don't already have a knife and some glue, you'd still be able to get everything you need for twelve builds and have change from £15. That's a lot of making for your pound.

VERDICT

Great fun, but we'd like the ability to carry on

9/10

Arduino Giga R1 WiFi

A big development board with huge ambitions

ARDUINO | **€68,70** | **hsmag.cc/giga**

By Ben Everard @ben_everard

The new Arduino Giga WiFi Rev 1 is the latest board from Arduino in the Mega form factor. This extends the classic Uno form with extra I/O options. It's a bit longer, but allows Arduino to pack a lot more onto the PCB.

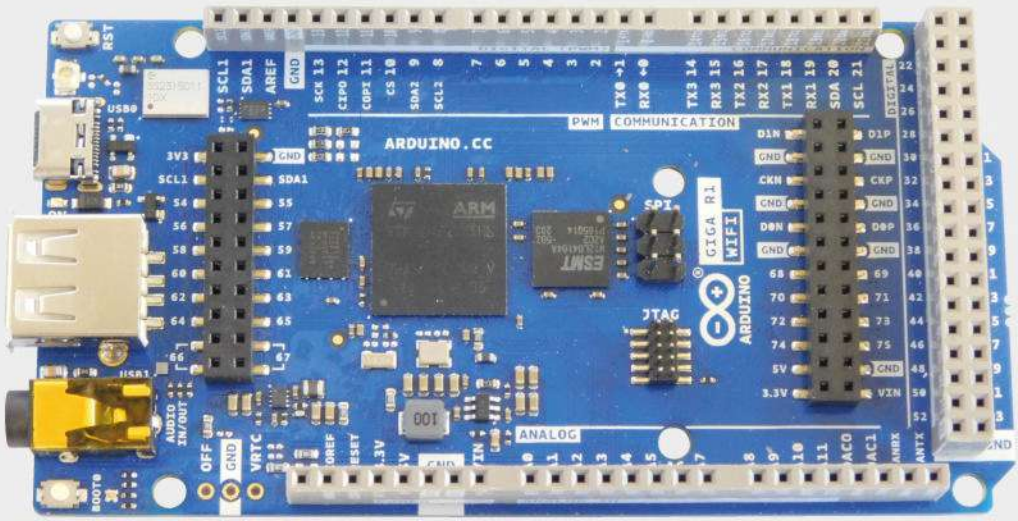
Firstly, there are a lot of pins – 76 of them. Honestly, we're not completely sure what to do with 76 pins, but if you need them, they're there. You could – if you so wished – control 5775 non-addressable LEDs, or – for a more colourful look – 1925 RGB LEDs.

Pins that don't fit into the old-school Mega format are available both above and below the board, so you can still access them if you put a Mega shield on it. It's not just the sheer number of pins either: there's support for quite a lot of different ways of attaching hardware. There's four UARTs, three I2Cs, two SPIs, CAN, an Arducam header, and MIPI display interface with support for up to 1024×768 pixels. It's not all digital, either. There are 16 analogue inputs and two digital outputs (available both in header pins and an audio jack). If wires aren't your thing, there's also support for Wi-Fi and Bluetooth, so you can send your data across the airwaves.

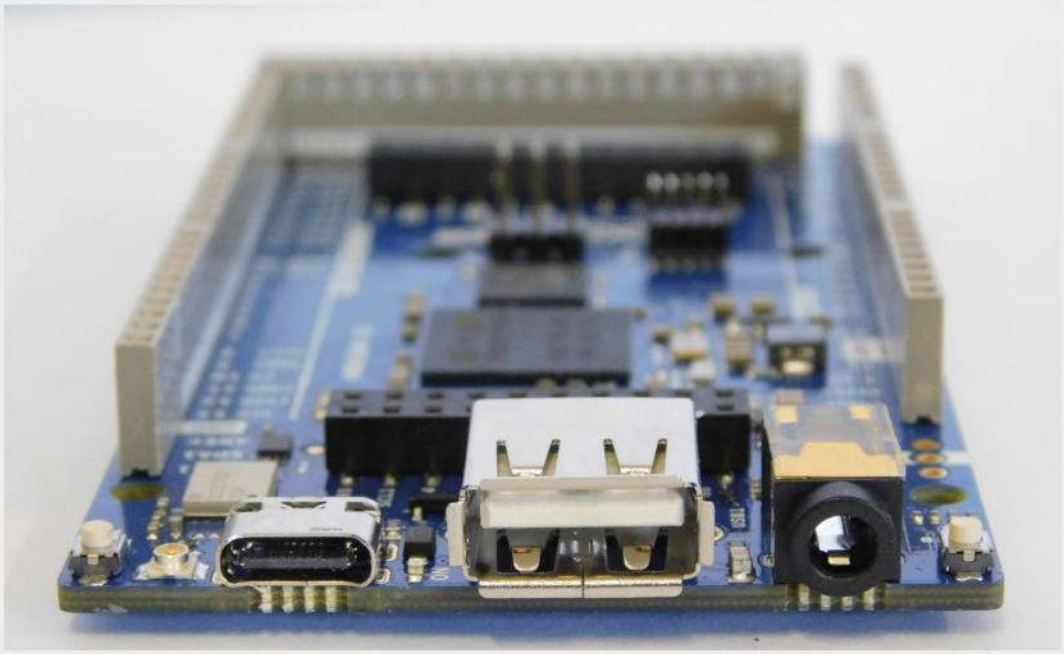
If you're going to drive all this I/O, you'll need a powerful processor, and the Giga doesn't disappoint. The STM32H747XI microcontroller has two processing cores – an Arm Cortex-M7 running at 480MHz, and an Arm Cortex-M4 running at 240MHz. You can upload sketches to either of them, and communicate between them using the RPC API. This lets code running on one core call functions in sketches on the other core. It also creates a serial-like communication stream between the sketches running on each core.

This is, let's be honest now, a ridiculously large amount of processing power for a microcontroller, and very few projects need this much oomph. However, there are a few cases where this amount of power is needed, for example, machine learning, image processing, or complex audio processing. In many of these cases, it may be useful to split hardware control from data processing and assign one core to each of these tasks. This is simple enough to do in practice.

Additionally, there's a real-time clock (RTC) for keeping track of time, and it can be powered separately to keep time even when the main board is turned off.



Below
Even the Mega format doesn't have enough pins for the Giga



Left
The USB-C is for data while the main USB is to connect peripherals

This massive range of features also comes with a pretty massive price tag – €68.70 (£60 / $73).

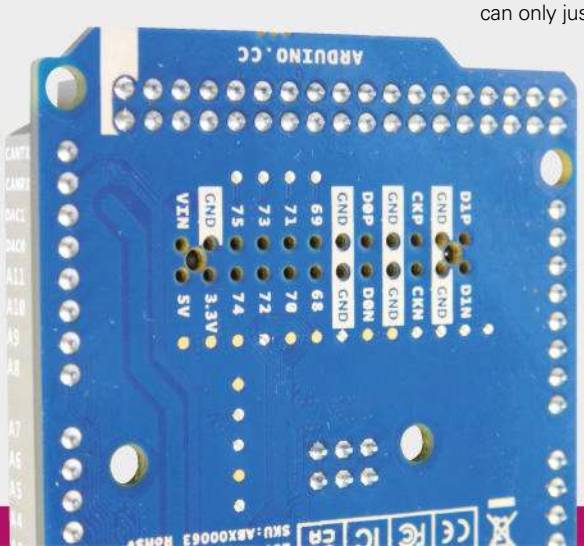
You can program this either in traditional Arduino sketches or via MicroPython. MicroPython is listed as a feature on the main product page, but if you dig deeper into the documentation, support is 'experimental'. It's a challenge to install, and we had some issues with it once it was up and running. At the time of writing, it's not in a state we could recommend it. This is a bit of a shame because it could be a great board for MicroPython – the raw processing power would be really useful in an interpreted language. Perhaps the support for this language will improve in the future.

In contrast to MicroPython, the Arduino IDE support feels much more complete. There are examples and a set of 'cheat sheets' that show you how to get started with the key features. If you're already familiar with the Arduino IDE, you'll get up and running with this board pretty quickly.

While Arduino obviously has a huge range of libraries for interfacing with other hardware, not all will work with the Giga. Some rely on specific hardware features. For example, at the time of writing, there was no library for WS2812B LEDs, aka NeoPixels. This isn't because the Giga can't work with them, it's just that someone needs to implement the protocol on this hardware. This shouldn't be an issue for much hardware, though, as anything using SPI, I2C, or other standard protocols should work.

> "A hugely powerful board that can crunch data at a **phenomenal rate and control a phenomenal amount of I/O**"

The Giga is a hugely powerful board that can crunch data at a phenomenal rate and control a phenomenal amount of I/O. There's no other hobbyist dev board quite like this available, and there are a whole range of projects that just became much easier. However, the high price means that this is a board that you can only justify for projects that actually need this power. We're looking forward to seeing what people can do with this.



Left
The additional pins are accessible from both the top and bottom of the board

VERDICT

Lots of power and lots of I/O cost lots of money.

9/10

CROWDFUNDING NOW

MNT Pocket Reform

A computer you can take apart

From $899 | **crowdsupply.com** | **Delivery:** October 2023

Laptops seem to have taken over the computing world. While this does mean that you can take your computing on the go, upgrades are typically very limited. This means that if you want a faster processor, you have to chuck away your keyboard, screen, battery, and everything else. Not only is this expensive, but it's also unnecessarily wasteful.

The MNT Reform is a laptop that bucks this trend. It's a portable computer that's easy to take apart to modify, or fix if something goes wrong. The MNT Pocket Reform is a new product in this line that's a small, ultra-portable machine.

It's powered by a quad-core Arm Cortex-A53 running at 1.8GHz with 8GB of RAM. It supports Debian Linux, so there's a wide variety of software available. While we've not tested one out, these specs should mean that it works well for light desktop use, but might struggle with processor-intensive tasks.

The hardware looks solid, but ultimately, it's the philosophy that will attract most people to this computer. Many of us grew up with computers that you could take apart and poke about inside, and would welcome the chance to get a bit of this hackability back in our portable machines.

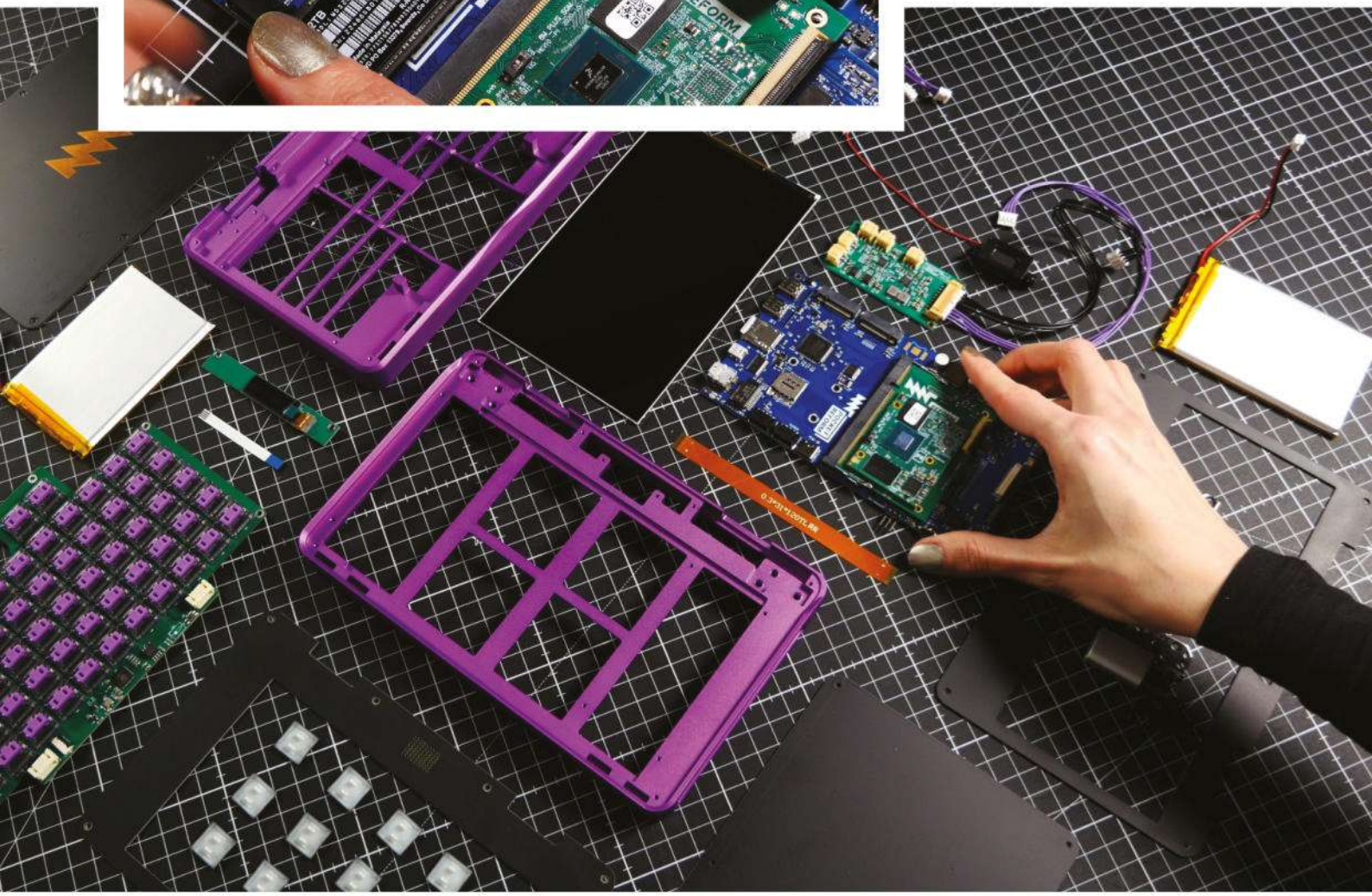




BUYER BEWARE

When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.





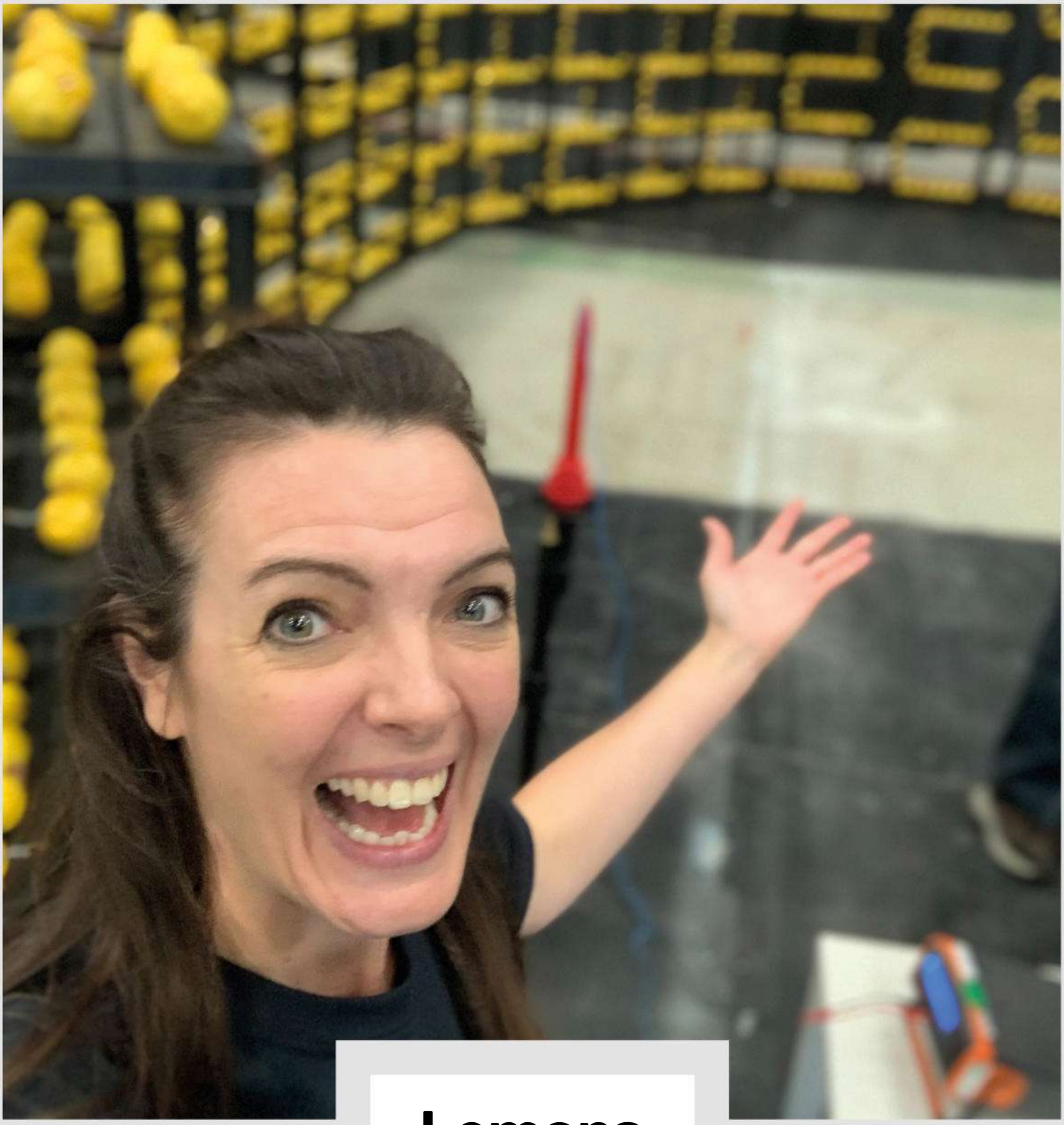
nextmen

issue
#67
ON SALE
18 MAY

photography
HACKS

ALSO
→ AMSTRAD
→ PRUSA MK4
→ RASPBERRY PI
→ AND MUCH MORE

DON'T MISS OUT
hsmag.cc/subscribe



Lemons

Anyone who can come up with a source of power that doesn't involve relying on despots (or burning liquified dinosaurs) deserves a medal. Or, in this case, a Guinness World Record. This is the world's most powerful fruit-powered battery comprising 2923 lemons and, in November 2021, it produced a Guinness-verified 2307.8 V of juice [sorry]. It was built for the Royal Society of Chemistry by a team of elite makers, including Fran Scott.

HiPi.io

PiKVM

Manage your servers or workstations remotely

A **cost-effective** solution for data-centers, IT departments or remote machines!



PiKVM HAT
for DIY and custom projects



Pre-Assembled version

- Real-time clock with rechargeable super capacitor
- OLED Display
- Bootable virtual CD-ROM & flash drive
- Serial console
- Open-source API & integration
- Open-source software

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz Váš dodavatel Raspberry Pi · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · ThePiHut · PiShop.us · CloudFree · PiShop.mx

Reseller suggestions and inquiries:
wholesale@hipi.io